AD-A069 770 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 9/2
DIAGNOSIS, SELF-DIAGNOSIS AND ROVING DIAGNOSIS IN DISTRIBUTED D--ETC(U)
SEP 78 R K NAIR DAAB07-72-C-0259
UNCLASSIFIED R-823 NL

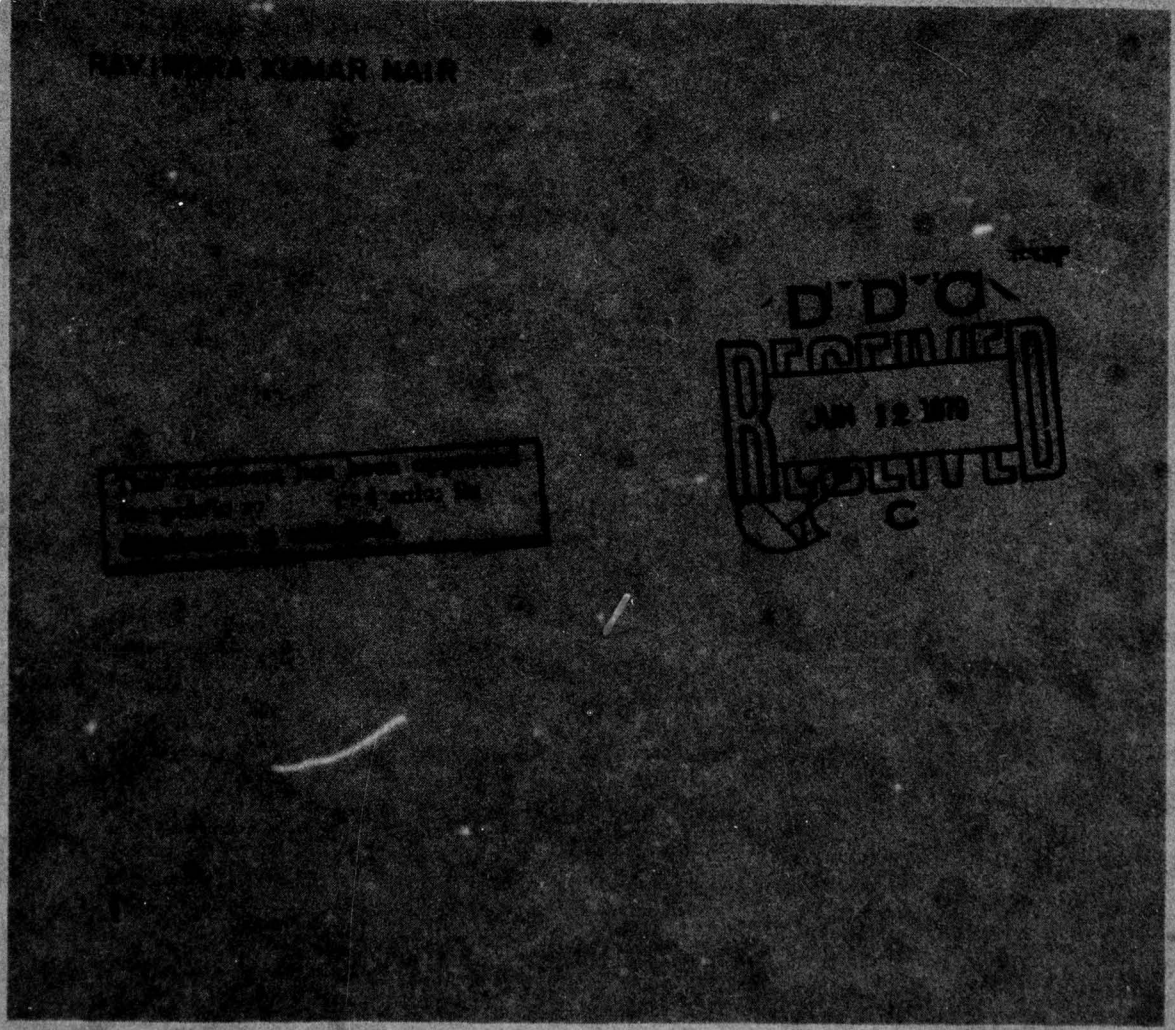
1 OF 2
AD A069770

REPORT R-823 SEPTEMBER, 1978 UILU-ENG 78-2216

CSL COORDINATED SCIENCE LABORATORY

DIAGNOSIS, SELF-DIAGNOSIS AND ROVING DIAGNOSIS IN DISTRIBUTED DIGITAL SYSTEMS

RAVINDRA KUMAR NAIR



UNIVERSITY OF ILLINOIS – URBANA, ILLINOIS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) (6) DIAGNOSIS, SELF-DIAGNOSIS AND ROVING DIAGNOSIS IN DISTRIBUTED DIGITAL SYSTEMS.	5. TYPE OF REPORT & PERIOD COVERED (9) Technical Report.
	6. PERFORMING ORG. REPORT NUMBER (14) R-823; UILU-ENG 78-2216
7. AUTHOR(s) (10) Ravindra Kumar Nair	8. CONTRACT OR GRANT NUMBER(s) (15) DAAB-07-72-C-0259
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, Illinois 61801	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (12) 144p.
11. CONTROLLING OFFICE NAME AND ADDRESS (11) Joint Services Electronics Program	12. REPORT DATE Sept[redacted]78
	13. NUMBER OF PAGES 132
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Diagnosable distributed digital systems	Roving graph
Fault tolerant computing	System Diagnosis
Implication index	System decomposition
Open circuit diagnosability	System reconfigurability
Roving diagnosis	System reusability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report studies various problems related to the diagnosis of systems which may be conveniently partitioned into subsystems on modules capable of performing tests among themselves. Traditionally, system diagnosis has aimed at the detection of one or more faulty units in systems. In certain critical operations like large distributed systems, it is desirable to know precisely which units in the system are fault-free. The report first considers the problem of determining at least one fault-free unit in a system. Necessary and sufficient conditions, as well as more convenient sufficient conditions for solving the →

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED 097 700

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (continued)

problem have been obtained.

With a view to facilitating diagnosis of very large systems, a study has been made to predict the behavior of the diagnosability when systems have known diagnosabilities are interconnected to other systems.

The "roving diagnosis" scheme introduced in the report is an attempt to achieve self-diagnosis in distributed digital systems while involving as few units in diagnosis as possible at any given time. A "window" is associated with the subsystem composed of the testing and tested units. Diagnosis of the entire system is achieved by moving this window around the system. Related issues which are studied include reconfiguration of the system under a fault and the reusability of the system after reconfiguration. Certain properties of the system communications graph are shown to lead to desirable systems where the delay in reconfiguring around a fault is small.

UILU-ENG 78-2216

DIAGNOSIS, SELF-DIAGNOSIS AND ROVING DIAGNOSIS IN DISTRIBUTED DIGITAL SYSTEMS

by

Ravindra Kumar Nair

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under Contrace DAAB-07-72-C-0259.

DIAGNOSIS, SELF-DIAGNOSIS AND ROVING DIAGNOSIS IN DISTRIBUTED DIGITAL SYSTEMS

Ravindra Kumar Nair, Ph.D.
Coordinated Science Laboratory and Department of Computer Science
University of Illinois at Urbana-Champaign, 1978

This thesis studies various problems related to the diagnosis of systems which may be conveniently partitioned into subsystems on modules capable of performing tests among themselves. Traditionally, system diagnosis has aimed at the detection of one or more faulty units in systems. In certain critical operations like large distributed systems, it is desirable to know precisely which units in the system are fault-free. The thesis first considers the problem of determining at least one fault-free unit in a system. Necessary and sufficient conditions, as well as more convenient sufficient conditions for solving the problem have been obtained.

With a view to facilitating diagnosis of very large systems, a study has been made to predict the behavior of the diagnosability when systems having known diagnosabilities are interconnected to other systems.

The "roving diagnosis" scheme introduced in the thesis is an attempt to achieve self-diagnosis in distributed digital systems while involving as few units in diagnosis as possible at any given time. A "window" is associated with the subsystem composed of the testing and tested units. Diagnosis of the entire system is achieved by moving this window around the system. Related issues which are studied include reconfiguration of the system under a fault and the reusability of the system after reconfiguration. Certain properties of the system communications graph are shown to lead to desirable systems where the delay in reconfiguring around a fault is small.

ACKNOWLEDGMENT

The author wishes to express his gratitude to Prof. Gernot Metze for his guidance, advice, support and encouragement at all times. The author also thanks Prof. Jacob Abraham for the many useful suggestions and discussions.

Special thanks are due to Mrs. June Wingler and Mrs. Mary McMillen for their excellent typing.

TABLE OF CONTENTS

Chapter Page

1. INTRODUCTION

1.1 System Diagnosis

Since the early days of research in diagnosis of digital systems much attention has been paid to the generation of test sets for the detection and location of component faults. The stuck-at-1 and stuck-at-0 model was the one most commonly used to solve the diagnosis problem by considering the logic behavior of the system under faulty and fault-free conditions. Results for the single stuck-at fault case have been very encouraging. Multiple stuck-at faults have been found to be more difficult to contend with.

In contrast to the component-level diagnosis as mentioned above, system-level diagnosis aims at a higher level, where the basic atoms are not gates but entire functional modules interconnected appropriately to form the system. A few reasons may be cited for the study of diagnosis at the system level. Firstly, the complexity of known multiple fault detection algorithms increases rapidly as the number of components involved increase. By breaking down the system into reasonable sized modules, the problem could be made more tractable. Secondly, functional modularization occurs naturally in present-day digital systems with the proliferation of large-scale integration. Consequently it restricts the range of interaction of faults and allows the diagnostician to take advantage of points within the system at which logic values are observable. It further serves as a convenient model for diagnosing distributed systems.

It must be mentioned here that system-level diagnosis is not an alternative to the traditional component-level diagnosis. System-level

diagnosis assumes that tests exist for faults in the modules composing the system. Also, once a faulty module is identified, component-level testing is needed to identify the faulty component, if it is appropriate for the technology at hand.

1.2 A Summary of Earlier Work

The pioneering work in the area of system diagnosis was done by Preparata, Metze and Chien [1]. The problem studied by them, termed the connection assignment problem, investigates the diagnosability of systems divided into subunits of convenient size and complexity. Each subunit in the system is assumed to be testable (completely, for some fault model) by some other subunit in the system. The result of a test is reliable only if the testing subunit is fault-free. Given the results of the tests performed by the various subunits and the interconnection between them, the problem of locating at least one (sequential diagnosability) or all (one-step diagnosability) of the faulty subunits is attacked. The assumption made is that no more than a specified number of subunits can become faulty between two testing periods. Further results in the area were obtained by Preparata [2], Hakimi and Amin [3] and Russell and Kime [4,5]. The work by Russell and Kime is noteworthy because their generalized results include cases where more than one unit is required to test a given unit. The probabilistic treatment of the connection assignment problem was undertaken by Maheshwari and Hakimi [6] and by Fujiwara and Kinoshita [7], while Mallela and Masson [8] studied the extension of the problem to the case of intermittent faults.

Barsi, Grandoni and Maestrini [9] introduced an interesting variation to the Preparata, Metze and Chien model by assuming that a

faulty unit will always be declared faulty by a good or a faulty unit. They show that the complexity of interconnections is reduced a good amount and that optimally connected networks are easier to find than in the original model.

In [10], Hayes proposes a technique for the design of fault tolerant systems. He associates a facility graph with every algorithm that is required to be executed by the system and proceeds to determine the existence of subgraphs which are isomorphic to the algorithm facility graph. Thus a system will be called k-fault-tolerant to a particular algorithm if for any fault involving k or fewer units, there exists a subgraph of good units which is isomorphic to the algorithm facility graph.

The graph-theoretic approach of Ramamoorthy [11], Ramamoorthy and Chang [12] and Ramamoorthy and Mayeda [13] forms the third type of work in system diagnosis. The concern here is primarily the possibility of identifying paths through the system such that the effect of the faults on the test stimuli at the system inputs may be observed at specified observation points termed the system outputs. To alleviate the problem of interference between units while diagnosing, they propose the use of test points and blocking gates [13]. This problem was pursued further by Batni and Kime [14] and, more recently, by Poisel and Kime [15].

1.3 Thesis Objectives

There are three directions in which further research in the area of system diagnosis may proceed:

(i) Just as in the case of component level diagnosis, the complexity of syndrome analysis in the case of system

level diagnosis increases rapidly as the number of nodes in the system graph increases. Further, the interconnection requirements make the system more inflexible when the number of faults that the system must tolerate increases. Thus it becomes necessary to reconsider the basic fault model for the system and attempt to solve the connection assignment problem for more realistic and possibly less restrictive fault models.

(ii) One of the major concerns in the maintenance of any system is the amount of time that is "lost" when the system is undergoing diagnosis. It appears that in a system composed of modules which can be tested by other modules, testing may be restricted to a small subset of modules while the rest of the system is busy doing useful computation. It should then be possible to devise a testing strategy which takes advantage of this potential parallelism, in order to improve performance through increase in available computational time.

(iii) Most systems which are deterministically designed to tolerate faults in a given fault model, are usually capable of tolerating many other unmodelled faults. In practice, there are a few critical faults in the fault model which the designed system can barely tolerate. However, these critical faults may be so improbable that, with a high degree of confidence, one may use the same system for more than the designed number of faults. In

order to perform theoretical analysis of such situations, the probabilities of various fault occurrences must be known.

This thesis takes the first two of the above mentioned three directions. Chapters 2 and 3 proceed as (i) above while a major part of the thesis, Chapters 4, 5 and 6, is devoted to a new testing strategy aiming at (ii) above.

In distributed systems, like large data bases, where for purposes of ensuring data validity the entire system may not be shut down at any time, it may be desirable to be able to pin-point at least one good unit in the system, instead of at least one faulty unit as in the sequential diagnosability problem of Preparata, Metze and Chien. Chapter 2 derives the necessary and sufficient conditions implied by such a system model, and compares the complexity implied by such systems with that of the known models.

Can large systems be diagnosed with reduced effort by analyzing syndromes for smaller subsystems within the system? With this question in mind, Chapter 3 investigates the behavior of composite systems when the subsystems composing the system have known diagnosabilities and specified characteristics.

The "roving diagnosis" scheme, introduced in Chapter 4, is an attempt to achieve self-diagnosis in distributed digital systems while involving as few units in diagnosis as possible at any given time. A "window" may be associated with the subsystem composed of the testing and tested units. Diagnosis of the entire system is achieved by moving this window around the system. Chapter 5 investigates the interconnections

and restrictions implied by this strategy, and presents techniques which aim at minimizing hardware external to the system. Chapter 6 employs the ideas of Chapter 5 in designing systems for given specifications. Chapter 6 also discusses how systems could reconfigure on the occurrence of a fault, what information needs to be transmitted to which nodes on the detection of a fault, and whether the reconfigured system is useful in further computation possibly with degradation in performance.

2. DETECTION OF FAULT-FREE UNITS IN SYSTEMS

2.1. Introduction

As described earlier, the connection assignment problem [1] considered the case when no more than a given number of faults can occur in a system, and when one subsystem alone is capable of testing another completely. A more general case was considered by Russell and Kime [4,5]. Instead of considering subsystems testing others, their model is formulated in terms of faults, tests and the relationships between them. A system is defined to be d-fault diagnosable with repair if and only if there exists a sequence of applications of a test set and repairs of identified faults that allows all faults originally present to be identified, provided the number of faults originally present does not exceed d [4]. Hence a system is d-fault diagnosable with repair if one application of the test set is sufficient to identify at least one fault present provided the number of faults originally present does not exceed d.

In certain applications, as in airline reservation systems, it may not be desirable to have any faulty units around, so that the integrity of the data base can be maintained. Under such circumstances, one step diagnosability [1] (also called d-fault diagnosability without repair [5]) may be the only acceptable solution. This condition has been shown to be very demanding in terms of nodes and interconnections required.

Another approach to the problem is to design for a system in which, after the application of a test sequence, at least m fault-free units can be identified (which can then proceed to execute tasks vital to the system) provided the number of faults present does not exceed a given number, say d. The properties of such systems will now be investigated.

2.2. Preliminaries

The model and notation will follow those of Russell and Kime [4] quite closely. Some of the details will be presented for completeness.

Associated with a system S there is assumed to be a set $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ of n possible faults and a set $\mathcal{T} = \{t_1, t_2, \ldots, t_p\}$ of p pass-fail tests. $F = \{F^1, F^2, \ldots, F^{2^n}\}$ is the set of all subsets of $\mathcal{F}$. $F(d)$ is the set of all fault patterns containing d or fewer faults and $F(d) \subseteq F$. The F-array for the system is a Boolean array having $F_i^k = 1$ if fault $f_i \in F^k$ and 0 otherwise.

A test t_j is a complete test for fault f_i if t_j always fails when f_i alone is present in S and always passes when the system is fault-free. The set of all complete tests for f_i is denoted by $t(f_i)$. Extending this notation, $t(f_i, f_j, \ldots, f_k) = t(f_i) \cup t(f_j) \cup \ldots \cup t(f_k)$.

$T(F^k)$ is a set of tests that are invalid in the presence of F^k. (A test t_j is invalid in the presence of fault pattern F^k if the presence of F^k causes the outcome of t_j to be unpredictable or unreliable.) The generalized fault array or the G-array for a system is defined as

$$G_j^k = \begin{cases} 0, & \text{if } t_j \notin t(F^k) \text{ and } t_j \notin T(F^k) \\ 1, & \text{if } t_j \in t(F^k) \text{ and } t_j \notin T(F^k) \\ X, & \text{otherwise.} \end{cases}$$

The diagnostic graph D of a system S is a labelled directed graph that has a vertex for each fault in $\mathcal{F}$ and a directed edge from the vertex associated with fault f_i to the vertex associated with fault f_j if and only if f_i invalidates at least one test that is complete for f_j, i.e., $T(f_i) \cap t(f_j) \neq \emptyset$. The directed edge is labelled by tests in the set

$T(f_i) \cap t(f_j)$. Directed edges to f_i with no origin are labelled by tests in the set $t(f_i) - T(\mathfrak{F})$.

Example 2.1

To illustrate these concepts, consider the system whose diagnostic graph is shown in Figure 2.1.

- $\mathfrak{F} = \{f_1, f_2, f_3, f_4\}$; $\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5, t_6\}$.
- The set of tests for the fault f_1 is $t(f_1) = \{t_4, t_5\}$.
- The set of faults that invalidate test t_4 is $\{f_2\}$.
- The set of faults that invalidate test t_5 is $\{f_3, f_4\}$.
- The set of tests invalidated by fault f_2 is $T(f_2) = \{t_2, t_3, t_4\}$.
- The set of faults invalidated by $\mathfrak{F}$ is $T(\mathfrak{F}) = T(f_1) \cup T(f_2) \cup T(f_3) \cup T(f_4) = \{t_1, t_2, t_3, t_4, t_5\}$.
- The set of tests that are not invalidated by any fault in $\mathfrak{F}$ is $\{t_6\}$.

□

2.3. Confirming the Absence of a Fault

In the Russell-Kime model, the determination of a fault-free unit is equivalent to the confirmation of the absence of a fault. This problem can be related to d/s diagnosability (see Friedman [16]).

A system S is said to be d/s diagnosable without repair (or one-step d/s diagnosable) if by a single application of the diagnostic test sequence any multiple fault $F^k \in F(d)$ can be diagnosed to within s single faults. The problem of confirming the absence of at least m faults under the assumption that there are no more than d faults in the system is then equivalent to the problem of d/(n - m) diagnosability without repair.

The cubical intersection operator $\sqcap$ as defined by Roth [17] will be used. Let $c^i, c^j \in \{0,1,X\}^n$. Then

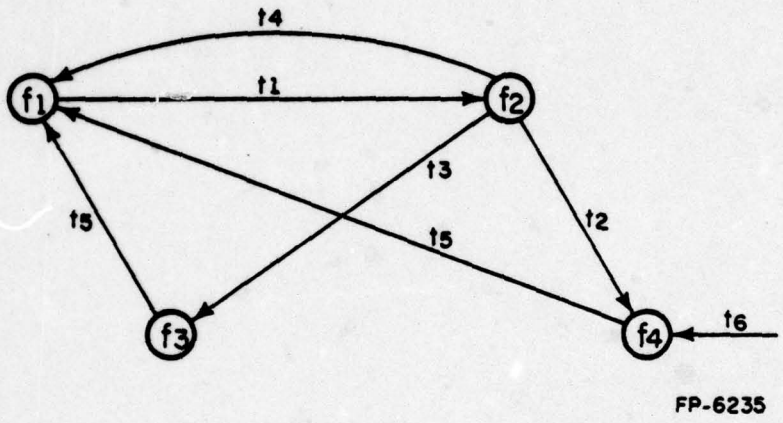


Figure 2.1 Diagnostic Graph for System of Example 2.1

$$c^i \sqcap c^j = \begin{cases} \emptyset \text{ (empty), if any } c^i_k \sqcap c^j_k = \emptyset \\ (c^i_1 \sqcap c^j_1 \times c^i_2 \sqcap c^j_2 \times \ldots \times c^i_n \sqcap c^j_n), \text{ otherwise} \end{cases}$$

with the $\sqcap$ operations among 0, 1 and X defined as in Table 2.1.

Table 2.1. Intersection Operation

$\sqcap$	0	1	X
0	0	$\emptyset$	0
1	$\emptyset$	1	1
X	0	1	X

Theorem 2.1

(a) A system $S = (\mathfrak{F},\mathcal{T},F,G)$ is d/s diagnosable without repair if and only if, for any $F^i,F^j,\ldots,F^k \in F(d)$, $|F^i \cup F^j \cup \ldots \cup F^k| > s$ implies $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$, $s < n$.

(b) A system $S = (\mathfrak{F},\mathcal{T},F,G)$ is d/n diagnosable without repair if and only if, for $F^k \in F(d)$, $F^k \neq \emptyset$ implies $G^k_j = 1$ for some $t_j \in \mathcal{T}$.

Proof:

(a) Let $|F^i \cup F^j \cup \ldots \cup F^k| > s$ imply $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$. Then $G^i \sqcap G^j \sqcap \ldots \sqcap G^k \neq 0$ implies $|F^i \cup F^j \cup \ldots \cup F^k| \leq s$. Hence after each application of the test sequence an outcome results that could have been due to the presence of any of the fault patterns in a set $\{F^i,F^j,\ldots,F^k\} \subseteq F(d)$. Since $|F^i \cup F^j \cup \ldots \cup F^k| \leq s$, the actual faults must be contained in a set of faults whose cardinality is not greater than s.

To prove necessity, let $G^i \sqcap G^j \sqcap \ldots \sqcap G^k \neq \emptyset$. Then each of $F^i,F^j,\ldots,F^k$ can give rise to the same test outcome. If, in addition, $|F^i \cup F^j \cup \ldots F^k| > s$, then at least $s+1$ possible faults must be considered to ensure inclusion of all the actual faults.

(b) S is d/n diagnosable without repair if and only if S is d-fault detectable, i.e., if and only if some test definitely fails for fault pattern $F^k \in \{F(d) - \emptyset\}$, i.e., if and only if, for some $t_j \in \mathcal{T}$, $G_j^k = 1$.

Q.E.D.

Example 2.2

Consider F(2) for the single-loop system whose graph is shown in Figure 2.2. Here $\mathfrak{F} = \{f_1, f_2, f_3, f_4, f_5\}$; $\mathcal{T} = \{t_1, t_2, t_3, t_4, t_5\}$; $F = F(2) = \{F^{12}, F^{13}, F^{14}, F^{15}, F^{23}, F^{24}, F^{25}, F^{34}, F^{35}, F^{45}\} \cup \{\mathfrak{F}\}$, where $F^{ij} = \{f_i, f_j\}$. The G-array for the system is shown in Table 2.2.

Table 2.2. G-Array for the Five-Node Single Loop System of Example 2.2

	t_1	t_2	t_3	t_4	t_5
f_1	1	X	0	0	0
f_2	0	1	X	0	0
f_3	0	0	1	X	0
f_4	0	0	0	1	X
f_5	X	0	0	0	1
F^{12}	1	X	X	0	0
F^{13}	1	X	1	X	0
F^{14}	1	X	0	1	X
F^{15}	X	X	0	0	1
F^{23}	0	1	X	X	0
F^{24}	0	1	X	1	X
F^{25}	X	1	X	0	1
F^{34}	0	0	1	X	X
F^{35}	X	0	1	X	1
F^{45}	X	0	0	1	X

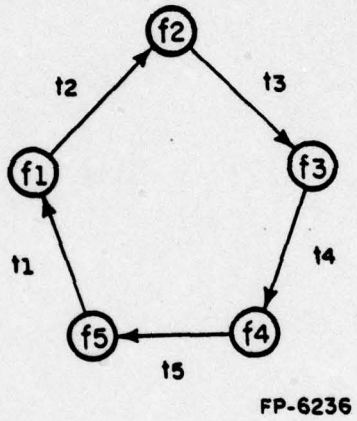


Figure 2.2 Diagnostic Graph for a Five-Node Single-Loop System

Now consider the syndrome $\{t_1,t_2,t_3,t_4,t_5\} = \{00101\}$. This syndrome could be produced by the following faults: F^{34} and F^{35}. $F^{34} \cap F^{35} = \{f_3\}$ and $G^{34} \sqcap G^{35} = 001 \times 1$. In this manner, one could verify for all faults $F^i,F^j,\ldots,F^k \in F(2)$ that $G^i \sqcap G^j \sqcap \ldots \sqcap G^k \neq \emptyset$ implies $|F^i \cup F^j \cup \ldots \cup F^k| \leq 3$. Hence the system is 2/3 diagnosable without repair.

□

While Theorem 2.1 does give the necessary and sufficient conditions for d/s diagnosability it is rather cumbersome to use in actual practice.

An observation that is clear from Theorem 2.1 is that a system which is d/s diagnosable without repair is also d/s' diagnosable without repair for $s' > s$, as long as $s' \leq n$. Hence we can define a system to be <u>s-critically d/s diagnosable without repair</u> if it is d/s diagnosable without repair but not d/(s - 1) diagnosable without repair.

<u>Example 2.3</u>

The five-node single-loop system of Example 2.2 may be verified to be s-critically 2/3 diagnosable without repair. □

<u>Theorem 2.2</u>

A system $S = (\mathcal{F},\mathcal{T},F,G)$ whose diagnostic graph is strongly connected can be s-critically d/(n - 1) diagnosable without repair only if for some test, the number of faults that invalidate the test is at least two.

<u>Proof</u>:

Assume otherwise. Let $\{F^1,F^2,\ldots,F^k\} \subseteq F(d)$ be a set of faults such that $|F^1 \cup F^2 \cup \ldots \cup F^k| = n - 1$ and $G^1 \sqcap G^2 \sqcap \ldots \sqcap G^k \neq \emptyset$. Such a set exists because S is s-critically d/(n - 1) diagnosable without repair. Consider f_x, F^g, F^h such that $f_x = \mathcal{F} - \{F^1 \cup F^2 \cup \ldots \cup F^k\}$, F^g is the set of all faults f_i such that $T(f_i) \cap t(f_x) \neq \emptyset$ and F^h is the set of all faults f_j

such that $T(f_x) \cap t(f_j) \neq \emptyset$. These faults exist because of the strongly connected nature of the diagnostic graph. Clearly $F^g, F^h \in \{F^1 \cup F^2 \cup \ldots \cup F^k\}$. $F^h \in \{F^1 \cap F^2 \cap \ldots \cap F^k\}$ because $f_x \notin \{F^1 \cup F^2 \cup \ldots \cup F^k\}$ and $T(f_x) \subseteq t(F^h)$. For all faults $F^p \in \{F^1, F^2, \ldots, F^k\}$ and all tests $t_x \in t(f_x)$, $G_x^p = 0$ or X, because $f_x \notin F^p$ while for all tests $t_y \in T(f_x)$, $G_y^p = 1$. If $G_x^p = X$, then consider the fault $F^{p'} = F^p \cup f_x$. Hence $G_x^{p'} = X$ and $G_y^{p'} = X$, implying that $G^1 \sqcap G^2 \sqcap \ldots \sqcap G^k \sqcap G^{p'} \neq \emptyset$ and $|F^1 \cup F^2 \cup \ldots \cup F^k \cup F^{p'}| > n - 1$, contradicting the assumption that the system is $d/(n-1)$ diagnosable without repair. Hence G_x^p must be 0 for all $F^d \in \{F^1, F^2, \ldots, F^k\}$ indicating that $|F^1 \cup F^2 \cup \ldots \cup F^k| < n - 1$, a contradiction. Q.E.D.

It is convenient to define a <u>single-mask-per-test system</u> (SMPT system) as one in which $T(f_i) \cap T(f_j) = \emptyset$ for all $f_i, f_j \in \mathfrak{F}$, $f_i \neq f_j$. A <u>strongly connected SMPT system</u> is one whose diagnostic graph is strongly connected. Hence the following corollaries can be stated.

<u>Corollary 2.2.1</u>: If a strongly connected SMPT system is $d/(n-1)$ diagnosable without repair then it must be $d/(n-2)$ diagnosable without repair.

<u>Corollary 2.2.2</u>: If a strongly connected SMPT system is $d/(n-m)$ diagnosable without repair, $m > 0$, then the system is d-fault diagnosable with repair (or sequentially d-fault diagnosable).

<u>Proof</u>:

Let $\{F^1, F^2, \ldots, F^k\} \subseteq F(d)$ be a set of faults such that $|F^1 \cup F^2 \cup \ldots \cup F^k| = n - m$ and $G^1 \sqcap G^2 \sqcap \ldots \sqcap G^k \neq \emptyset$. Let f_x, f_y be faults such that $f_x \in \mathfrak{F} - \{F^1 \cup F^2 \cup \ldots \cup F^k\}$, $f_y \in \{F^1 \cup F^2 \cup \ldots \cup F^k\}$ and $T(f_x) \cap t(f_y) \neq \emptyset$. Let $t_y \in T(f_x) \cap t(f_y)$. Then for all faults $F^p \in \{F^1, F^2, \ldots, F^k\}$, $G_y^p = 1$, indicating that $f_y \in \{F^1 \cap F^2 \cap \ldots \cap F^k\}$. Hence the system is sequentially d-fault diagnosable. Q.E.D.

Lemma 2.3

If a strongly connected SMPT system is sequentially d-fault diagnosable, then it must be d/(n - m) diagnosable without repair for some m > 0.

Proof:

Let $\{F^1,F^2,\ldots,F^k\} \subseteq F(d)$ be a set of faults such that $F^1 \cap F^2 \cap \ldots \cap F^k \neq \emptyset$ and $G^1 \sqcap G^2 \sqcap \ldots \sqcap G^k \neq \emptyset$. Such a set must exist because the system is sequentially d-fault diagnosable. Let $f_x \in F^1 \cap F^2 \cap \ldots \cap F^k$. Clearly for all tests $t_x \in t(f_x)$, $G_x^p = 1$ or X for $F^p \in \{F^1,F^2,\ldots,F^k\}$. If $G_x^p = X$ then for the fault $F^p - f_x = F^{p'}$, $G_x^{p'} = 1$ or X and for all tests $t_y \in T(f_x)$ $G_y^p = X$, $G_y^{p'} = 0$ or 1. Hence $G^1 \sqcap G^2 \sqcap \ldots \sqcap G^k \sqcap G^{p'} \neq \emptyset$, but $F^1 \cap F^2 \cap \ldots \cap F^k \cap F^{p'} \subset F^1 \cap F^2 \cap \ldots \cap F^k$. Proceeding in this manner for other faults belonging to $F^1 \cap F^2 \cap \ldots \cap F^k$, faults $F^{q'},\ldots,F^{v'}$ may be found so that $G^1 \sqcap G^2 \sqcap \ldots \sqcap G^k \sqcap G^{p'} \sqcap G^{q'} \sqcap \ldots \sqcap G^{v'} \neq \emptyset$, but $F^1 \cap F^2 \cap \ldots \cap F^k \cap F^{p'} \cap F^{q'} \cap \ldots \cap F^{v'} = \emptyset$ violating the sequential d-fault diagnosability of the system. Hence there must be some $f_x \in F^1 \cap F^2 \cap \ldots \cap F^k$ such that $G_x^p = 1$ for all $F^p \in \{F^1,F^2,\ldots,F^k\}$, indicating that $|F^1 \cup F^2 \cup \ldots \cup F^k| < n$. Q.E.D.

Theorem 2.3

A strongly connected SMPT system is d/(n - m) diagnosable without repair for some m > 0 if and only if it is sequentially d-fault diagnosable with repair.

Proof:

Follows from the proofs of Corollary 2.2.2 and Lemma 2.3.

The above results are particularly useful for the systems that have been studied traditionally like the single-loop systems or the $D_{\delta A}$ systems [1]. All these studies assume that a subsystem or unit can be

tested completely by another unit. This would imply that when more units are employed to test some unit each of the units performs a complete test, independent of the other testing units. Such systems are hence SMPT systems. Most of the systems studied also tend to be strongly connected, although there appears to be no reason for this to be necessarily true in practice.

Theorem 2.4

A connected SMPT system is sequentially d-fault diagnosable only if it is $d/(n-m)$ diagnosable without repair for some $m > 0$.

Proof:

Since the system is connected every fault must have some test which is invalidated by another fault. However, there may exist faults which do not invalidate any test. The proof then is exactly the same as that for Lemma 2.3.

It is not difficult to obtain a counterexample for the converse of Theorem 2.4. Thus sequential d-fault diagnosability is a sufficient, but not necessary, condition for $d/(n-1)$ diagnosability in a system. This means then that in systems where one unit can be tested completely by another, the problem of finding one good unit is never more difficult than that of finding one bad unit.

Example 2.4

The five-node single-loop system of Example 2.2 is an SMPT system because there does not exist a set of two faults $\{f_i, f_j\}$ such that $f_i \neq f_j$ and $T(f_i) \cap T(f_j) \neq \emptyset$. It was mentioned that this system is 2/3 diagnosable with repair. Since $n = 5$, by Theorem 2.3, $m > 0$ and hence the system must

be sequentially 2-fault diagnosable with repair. This is indeed the case, as seen from Theorem 5 in the paper by Preparata, Metze and Chien [1].

□

Example 2.5

Consider a general n-node single loop system. If n satisfies the condition that $n > (m+1)^2 + \lambda(m+1)$ for some integer value of m and $\lambda = 0,1$ then Preparata [2] has shown that the diagnosability with repair of the system must be at least $d = 2m + \lambda$. Let $\mathcal{F} = \{f_0, f_1, \ldots, f_{n-1}\}$, $\mathcal{T} = \{t_0, t_1, \ldots, t_{n-1}\}$, $t(f_i) = t_i$ and $T(f_i) = t_{i+1 \bmod n}$, $i = 0,1,\ldots,n-1$. Then a syndrome can be expressed as a vector where each element is a 0 or a 1 and the ith element corresponds to the result of test i.

Consider any syndrome. The (n - 1)th element will be considered adjacent to the 0th element. It can be shown that the length of the largest consecutive string of 0's in any valid syndrome assuming no more than d faults must be at least m+1. If the tests corresponding to this string are $\{t_p, t_{p+1}, \ldots, t_{q-1}, t_q\}$ where $q - p \bmod n \geq m$ then the fault f_q must be absent or else there would be at least d+1 faults. Hence the system is at least d/(n - 1) diagnosable with repair, as suggested by Theorem 2.3.

□

Example 2.6

The following example (Figure 2.3) shows that a system is d/(n-m) diagnosable without repair is not necessarily sequentially d-fault diagnosable.

It may be verified that the syndrome {0101010} as the outcome of tests $\{t_1, t_2, \ldots\ldots, t_6, t_7\}$ can result only from the absence of fault f_7 if no more than 3 faults can occur simultaneously. The system is however not sequentially 3-fault diagnosable.

□

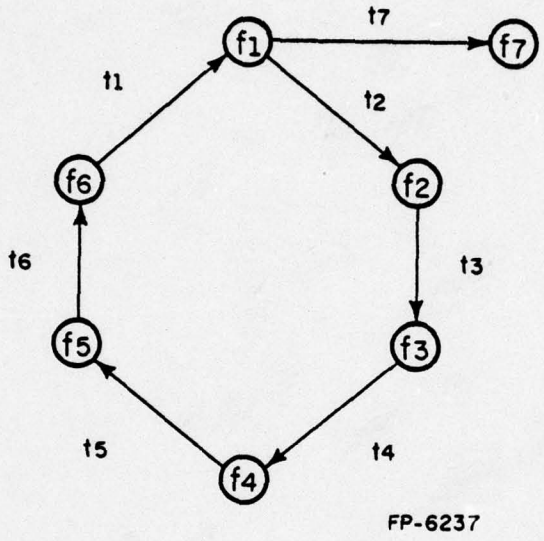


Figure 2.3 A Connected SMPT System Diagnostic Graph for Example 2.6

A <u>subsystem</u> $S' = (\mathfrak{F}', \mathcal{T}', F', G')$ of a system $S = (\mathfrak{F}, \mathcal{T}, F, G)$ is defined as a system in which

(i) $\mathfrak{F}' \subseteq \mathfrak{F}$, $\mathcal{T}' \subseteq \mathcal{T}$, $F' \subseteq F$, $G' \subseteq G$, and

(ii) if $t_i \in \mathcal{T}'$, then $f_i, f_j \in \mathfrak{F}'$ where $t_i \in t(f_i)$ and $t_j \in T(f_i)$, i.e., for every test from S that is included in S', the fault for which it is a test and all the faults that invalidate the test must be included in S' also.

<u>Theorem 2.5</u>

A system $S = (\mathfrak{F}, \mathcal{T}, F, G)$ is $d/(n-1)$ diagnosable without repair if and only if there exists a subsystem $S' = (\mathfrak{F}', \mathcal{T}', F', G')$ of S with $|\mathfrak{F}'| = z$, such that S' is $(d-n+z)/(z-1)$ diagnosable without repair.

<u>Proof</u>:

If such a subsystem exists then by simply considering the subset $\mathcal{T}'$ of tests in $\mathcal{T}$, the absence of a fault can be ascertained because the assumption of at most $d-(n-z)$ faults in S' implies at most d faults in S. Conversely, assume that there does not exist a subsystem which is $(d-n+z)/(z-1)$ diagnosable. Then there exist faults $\{F^i, F^j\} \subseteq F(d-n+z)$ such that $|F^i \cup F^j| > z-1$, $G^i \sqcap G^j \neq \emptyset$ and $\{F^i, F^j\} \subseteq F'(d-n+z)$ for some subsystem S' of S. Consider the faults $F^p = F^i \cup \{\mathfrak{F} - \mathfrak{F}'\}$ and $F^q = F^j \cup \{\mathfrak{F} - \mathfrak{F}'\}$. Clearly $G^p_g = G^q_g = X$ for all tests $t_g \in T(\mathfrak{F} - \mathfrak{F}')$. Since $|\mathfrak{F} - \mathfrak{F}'| = n - z$, $|F^p \cup F^q| > n-1$, but $G^p \sqcap G^q \neq \emptyset$, contradicting the assumption that S is $d/(n-1)$ diagnosable. Q.E.D.

The necessity of the condition in Theorem 2.5 implies that to build a system in which a good unit can always be found, under the assumption that no more than d faults can occur, there must exist a "kernel" in the system in which a good unit can always be found assuming that the

number of faults is no more than a certain maximum, which depends on the number of units in the kernel.

A <u>self-testing system</u> $S = (\mathfrak{F},\mathcal{T},F,G)$ is defined as one in which $T(\mathfrak{F}) = t(\mathfrak{F})$, i.e., every test in the system is invalidated by some fault in the system.

<u>Theorem 2.6</u>

A necessary condition for a self-testing system $S = (\mathfrak{F},\mathcal{T},F,G)$ to be $d/(n-1)$ diagnosable without repair is that $n \geq 2d + 1$.

<u>Proof</u>:

Let $n \leq 2d$. Then consider a fault F^k such that $|F^k| = n$. Let F^i, F^j be nonempty disjoint fault patterns such that $F^i, F^j \in F(d)$ and $F^i \cup F^j = F^k$. Suppose some $G_h^i = 1$. Since $t_h \in t(F^i)$ and $t_h \notin T(F^i)$, it must be that $t_h \in T(F^j)$ from the self-testing nature of the system. Hence $G_h^j = X$. Similarly $G_h^j = 1$ implies $G_h^i = X$. This means that $G^i \sqcap G^j \neq \emptyset$ in spite of the fact that $|F^i \cup F^j| > n - 1$. Hence the system is not $d/(n-1)$ diagnosable without repair. Q.E.D.

For a fault pattern F^k, the <u>complement</u> of the pattern $\overline{F^k}$ is defined as the set of faults $\mathfrak{F} - \{F^k\}$. A fault pattern F^k is said to be a <u>masking pattern</u> if and only if every complete test in $\mathcal{T}$ for each fault in $\{\overline{F^k}\}$ is invalid in the presence of F^k, i.e., $t(\overline{F^k}) \subseteq T(F^k)$. The cardinality of the smallest masking pattern in F is termed the <u>system masking index</u>, denoted by M(S).

<u>Theorem 2.7</u>

For a system S to be $d/(n-1)$ diagnosable without repair, $d \leq M(S) \leq n$.

Proof:

If $\{F^p\} = \mathfrak{F}$, then $\overline{F^p} = \emptyset$ and $t(\emptyset) = \emptyset \subseteq T(F^p)$. Hence $M(s) \leq n$ is obvious. Assume that $M(S) < d$. Then there exists a fault $F^q \in F(d-1)$ such that $t(\overline{F^q}) \subseteq T(F^q)$. Consider the faults $F^i = F^q \cup f_1$, $F^j = F^q \cup f_2, \ldots$, $F^k = F^q \cup f_r$, where $\overline{F^q} = \{f_1, f_2, \ldots, f_r\}$. Clearly $F^i, F^j, \ldots, F^k \in F(d)$ because $F^q \in F(d-1)$. Also $\overline{F^i} \cap \overline{F^j} \cap \ldots \cap \overline{F^k} = \emptyset$. For any test $t_x \in t(f_x)$, where $f_x \in \overline{F^q}$, $G^q_x = X$ because $t_x \in T(F^q)$. Hence $G^q_x \sqcap G^i_x \sqcap G^j_x \sqcap \ldots \sqcap G^k_x \neq \emptyset$ and $|F^q \cup F^i \cup \ldots \cup F^k| = n$. But this implies that the system is not $d/(n-1)$ diagnosable without repair. Q.E.D.

Lemma 2.8.1 [4]

A system $S = (\mathfrak{F}, \mathcal{T}, F, G)$ is d-fault diagnosable with repair if and only if, for any $F^i, F^j, \ldots, F^k \in F(d)$, $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ implies $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$.

Lemma 2.8.2 [4]

Let $\{F^i, F^j, \ldots, F^k\} \subseteq F(d)$ be any set of two or more fault patterns which is minimal with respect to the property that $F^i \cap F^j \cap \ldots F^k = \emptyset$. Then $|F^i \cup F^j \cup \ldots \cup F^k| \leq \lfloor (\frac{d+2}{2})^2 \rfloor$.

Theorem 2.8

A sufficient condition for a system S to be $d/(n-1)$ diagnosable without repair is that S be d-fault diagnosable with repair and $n \geq \lfloor (\frac{d+2}{2})^2 \rfloor$.

Proof:

Let $S = (\mathfrak{F}, \mathcal{T}, F, G)$ and let $\{F^g, F^h, \ldots, F^q\} \subseteq F(d)$ such that $|F^g \cup F^h \cup \ldots \cup F^q| = n$. Contained in that set is a subset of two or more faults $\{F^i, F^j, \ldots, F^k\}$ such that $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ (by Lemma 2.8.2). Since S is d-fault diagnosable with repair, by Lemma 2.8.1, $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$. Hence $G^g \sqcap G^h \sqcap \ldots \sqcap G^q = \emptyset$, satisfying Theorem 2.1(a). Q.E.D.

Example 2.7

This example shows that a system which is d-diagnosable with repair but for which $n < \lfloor (\frac{d+2}{2})^2 \rfloor$ is not necessarily $d/(n-1)$ diagnosable without repair. Consider the diagnostic graph of Figure 2.4. In this system, the fault f_4 can be detected unambiguously by test t_4, which cannot be invalidated. If f_4 is absent then t_1 will give a reliable result, while if f_1 is absent tests t_2 and t_3 cannot be invalidated. Thus at least one fault can always be detected (irrespective of the number of faults in the system). However a syndrome like (11111) cannot indicate the absence of a fault unambiguously, even if there can be no more than 3 faults simultaneously occurring in the system. □

A fault pattern F^k is closed if and only if every complete test in $\mathcal{T}$ for each fault in F^k is invalid in the presence of F^k [4], i.e., $t(F^k) \subseteq T(F^k)$. The implication I^j of a fault F^j is the cardinality of the smallest closed fault pattern in F containing the fault F^j. The system implication index, I(S), is defined as the cardinality of the largest implication for faults in F.

The system closure index [4] denoted by c(S) is the cardinality of the smallest closed fault pattern in F.

Lemma 2.8.3 [4]

A sufficient condition for a system S to be d-fault diagnosable with repair is that $c(S) > \lfloor (\frac{d+2}{2})^2 \rfloor$. For $d = n$, S is n-fault diagnosable with repair if and only if $c(S) = \infty$.

Corollary 2.8: A sufficient condition for a self-testing system S to be $d/(n-1)$ diagnosable without repair is that $c(S) > \lfloor (\frac{d+2}{2})^2 \rfloor$.

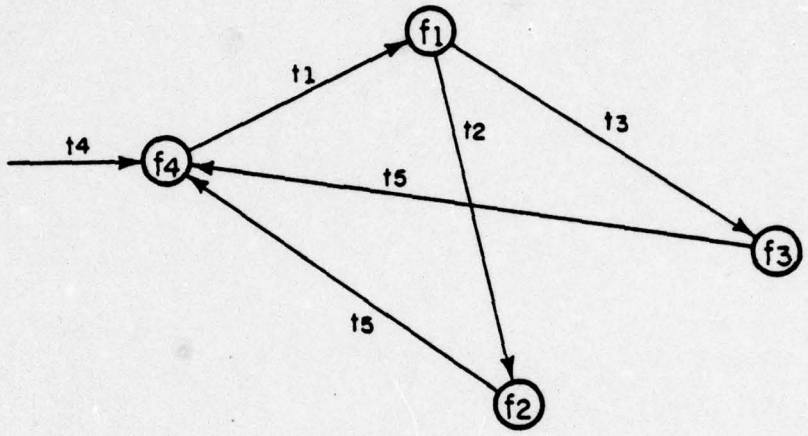


Figure 2.4 Diagnostic Graph of a System Which is 4-Fault Diagnosable with Repair

Proof:

By Lemma 2.8.3, the system must be d-fault diagnosable with repair. Since S is self-testing, $c(S) \leq n$, implying that $n > \lfloor(\frac{d+2}{2})^2\rfloor$. By Theorem 2.8 the system must hence be $d/(n-1)$ diagnosable without repair. Q.E.D.

Lemma 2.9.1

For a system $S = (\mathfrak{F},\mathcal{T},F,G)$ let $F^i,F^j,\ldots,F^k \in F$ be two or more fault patterns such that $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$, and $F^p = F^i \cup F^j \cup \ldots \cup F^k$. If $|F^p| < I^p$ then $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$.

Proof:

Since F^p is not closed, there exists some test $t_j \in t(F^p)$ such that $t_j \notin T(F^p)$. Let $t_j \in t(f_b)$. Then $f_b \in F^p$ and since $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$, there exist $F^g,F^h \in \{F^i,F^j,\ldots,F^k\}$ such that $f_b \in F^g$, $f_b \notin F^h$. Since $t_j \notin T(F^g)$ and $t_j \notin T(F^h)$, $G^g \sqcap G^h = \emptyset$. Q.E.D.

A single test per fault (STPF) system is one in which every fault has one and only one test, i.e., $|t(f_i)| = 1$ for every $f_i \in \mathfrak{F}$.

Theorem 2.9

For a connected STPF system S, a sufficient condition for $d/(n-1)$ diagnosability without repair is that $I(S) > \lfloor(\frac{d+2}{2})^2\rfloor$.

Proof:

Consider a set of faults $\{F^g,F^h,\ldots,F^q\} \subseteq F(d)$ such that $|F^g \cup F^h \cup \ldots \cup F^q| = n$. Since $I(S) > \lfloor(\frac{d+2}{2})^2\rfloor$ there must exist a fault $f_p \in \mathfrak{F}$ such that $I(f_p) = I(S) > \lfloor(\frac{d+2}{2})^2\rfloor$. Hence there must exist at least j faults $f_1,f_2,\ldots,f_j \in \mathfrak{F}$ such that $I(f_1) = I(f_p) = I(S)$, $I(f_2) \geq I(S) - 1$, $I(f_3) \geq I(S) - 2$, ..., $I(f_j) \geq I(S) - j + 1$, $j \leq I(S)$. Since $I(f_1) > \lfloor(\frac{d+2}{2})^2\rfloor$, it must also be the case that $I(f_2) > \lfloor(\frac{(d-1)+2}{2})^2\rfloor$, $I(f_3) > \lfloor(\frac{(d-2)+2}{2})^2\rfloor$, ..., $I(f_{d+1}) > \lfloor(\frac{(d-d)+2}{2})^2\rfloor$. Now let $\{F^i,F^j,\ldots,F^k\}$ be a

subset of faults in $\{F^g, F^h, \ldots, F^q\}$ which is minimal with respect to the property $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ and $f_r \in F^i \cup F^j \cup \ldots \cup F^k$, $f_l \in F^g \cap F^h \cap \ldots \cap F^q$ for $i > r$. There must exist such a fault f_r (because $|F^g \cap F^h \cap \ldots \cap F^q| > d$ otherwise). Clearly, $|F^i \cup F^j \cup \ldots \cup F^k| \leq \lfloor (\frac{d-r+1+2}{2})^2 \rfloor$ by Lemma 2.8.2. $I(F^i, F^j, \ldots, F^k) \geq I(f_r) > \lfloor (\frac{(d-r+1)+2}{2})^2 \rfloor$. Hence $|F^i \cup F^j \cup \ldots \cup F^k| > I(F^i, F^j, \ldots, F^k)$, which by Lemma 2.9.1 implies that $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$. This in turn implies $G^g \sqcap G^h \sqcap \ldots \sqcap G^q = \emptyset$ and hence the system S must be $d/(n-1)$ diagnosable without repair. Q.E.D.

Example 2.8

Reconsidering the system of Example 2.6, the diagnostic graph for which is shown in Figure 2.3, the following may be noted.

The implication for the various faults is as follows: $I(f_1) = 6$, $I(f_2) = 6$, $I(f_3) = 6$, $I(f_4) = 6$, $I(f_5) = 6$, $I(f_6) = 6$, $I(f_7) = 7$. Hence the system implication index is given by $I(S) = I(f_7) = 7$.

By Theorem 2.9, the system must be 3/6 diagnosable without repair, since $(\frac{d+2}{2})^2\big|_{d=3} = 6\frac{1}{4}$ and $I(S) = 7$.

On the other hand the system closure index is only 6 and this does not satisfy Russell and Kime's condition [4] that $c(s) > \lfloor (\frac{d+2}{2})^2 \rfloor$. Hence the system is not 3-fault diagnosable with repair. Both these results are in keeping with the observation in Example 2.6 that a syndrome for 3 faults exists for which the absence of some fault can be guaranteed but no fault can be unambiguously identified. □

2.4. Discussion

By using a general fault model first developed by Russell and Kime [4], the theory has been developed in this chapter for detecting the

absence of a fault in a system. In distributed systems, the detection of the absence of a fault is equivalent to the detection of a fault-free unit. The relationship between this problem and that of d-fault diagnosability with repair (or the problem of finding a faulty unit in the system) has been detailed.

The basic necessary and sufficient conditions for the confirmation of absence of a fault have been derived. With the help of a parameter termed the system implication index, more convenient sufficient conditions have also been developed.

3. DIAGNOSABILITY AND SYSTEM DECOMPOSITION

3.1. Introduction

The general techniques for determining diagnosabilities of systems with or without repair or for determining the faulty units given the syndrome are not very efficient. For particular cases, as in Karunanithi and Friedman [18] or Smith [19], algorithms may be found which take polynomially-bound time for execution. In most general cases, however, the complexity of the solution techniques becomes unpractically high as the number of system components becomes large.

One possible approach to the problem would be to break down the problem into smaller parts, so that the analysis of the parts can be performed in parallel. However, interaction between the parts must be considered in many cases. This can be done by considering the results from the various parts, rather than the detailed network configuration.

Two advantages result from using such a hierarchical approach. First, the ability to introduce parallelism in the algorithm implies more efficient algorithms for analysis. Second, the complexity of the global analyzer decreases because of the smaller problems which it now has to handle. This translates to a reduced "hard-core" requirement for purposes of diagnosis.

An approach in this direction was made by McPherson and Kime [20] who modified the Russell and Kime approach [4,5] so that each fault was identified with a "part" of the system. It was also assumed that no more than a certain number of faults could occur in a given "part" of the system.

This section will present some results which help in giving an idea of the diagnosability (with or without repair) of systems composed of subsystems of known diagnosabilities.

3.2. STPF and SMPT Systems

For STPF (single-test-per-fault) and SMPT (single-mask-per-test) systems, the problem of finding the diagnosability of composite systems, given subsystem diagnosabilities, is probably somewhat simpler than for the general case.

Lemma 3.1.1 (Russell and Kime [4])

An STPF system S is d-fault diagnosable with repair if and only if the closure index of the system $c(S) > \left\lfloor \left(\frac{d+2}{2}\right)^2 \right\rfloor$.

A composite system $S = (\mathcal{F},\mathcal{T},F,G)$ of two composing subsystems $S_A = (\mathcal{F}_A,\mathcal{T}_A,F_A,G_A)$ and $S_B = (\mathcal{F}_B,\mathcal{T}_B,F_B,G_B)$ is said to be that system in which $\mathcal{F} = \mathcal{F}_A \cup \mathcal{F}_B$, $\mathcal{T} \supseteq \mathcal{T}_A \cup \mathcal{T}_B$, $\{T(\mathcal{F}_A) - T_A(\mathcal{F}_A)\} \cap t(\mathcal{F}_A) = \emptyset$ and $\{T(\mathcal{F}_B) - T_B(\mathcal{F}_B)\} \cap t(\mathcal{F}_B) = \emptyset$. If D, D_A and D_B are the diagnostic graphs for S, S_A and S_B respectively then the last two conditions imply that the only edges in D which are not in either D_A or D_B are the ones between nodes in D_A and those in D_B.

Example 3.1

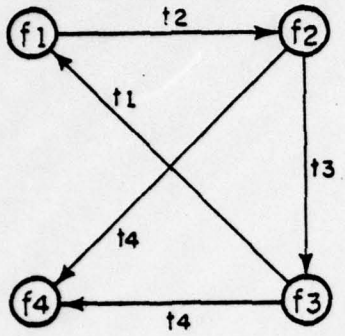
Figure 3.1 shows a sample composition of two subsystems, S_A and S_B. The diagnostic graphs for the two composing systems are depicted in Figures 3.1(a) and (b). The composite system whose diagnostic graph is shown in Figure 3.1(c) has interconnections in such a way that exactly two faults in each subsystem are invalidated by exactly two faults in the other subsystem. Thus

$$\mathcal{F}_A = \{f_1,f_2,f_3,f_4\},\ \mathcal{F}_B = \{f_5,f_6,f_7,f_8,f_9\};$$

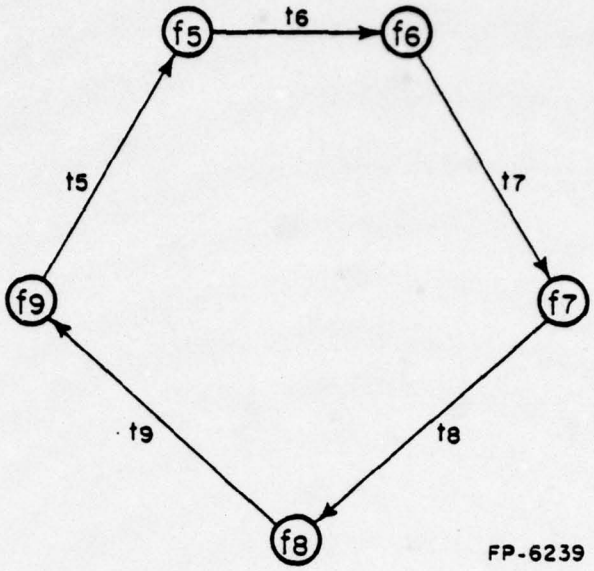
$$\mathcal{F} = \{f_1,f_2,f_3,f_4,f_5,f_6,f_7,f_8,f_9\};$$

$$\mathcal{T}_A = \{t_1,t_2,t_3,t_4\},\ \mathcal{T}_B = \{t_5,t_6,t_7,t_8,t_9\};$$

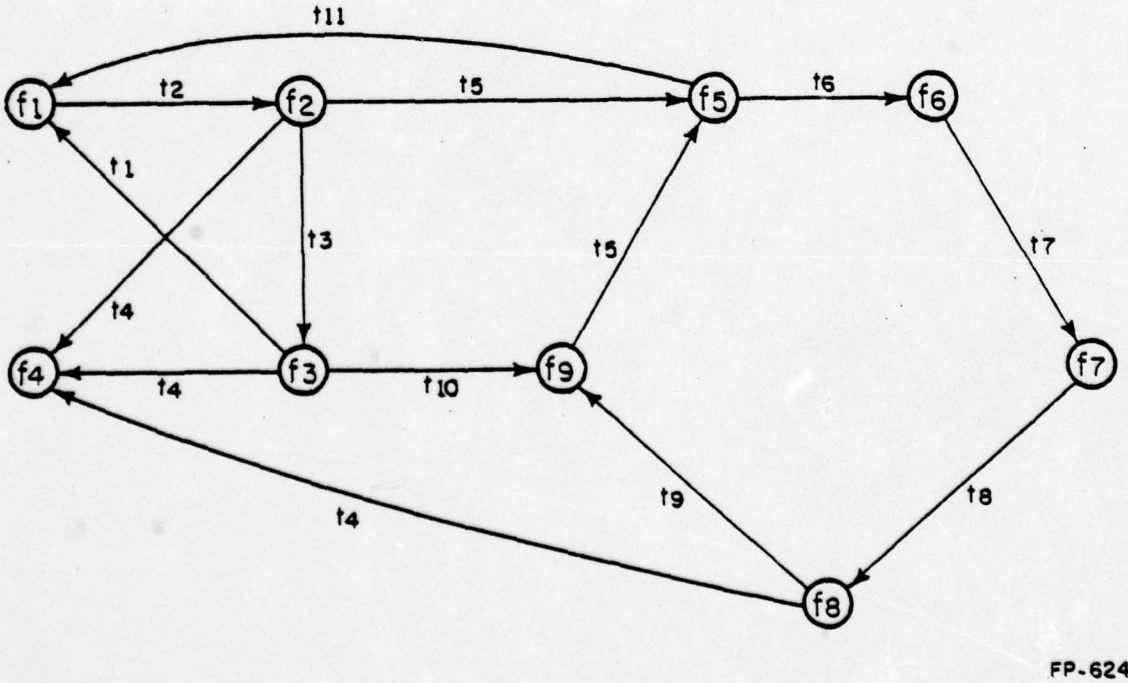
$$\mathcal{T} = \{t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_9,t_{10},t_{11}\}.$$



(a) Diagnostic Graph for Subsystem S_A



(b) Diagnostic Graph for Subsystem S_B



(c) Diagnostic Graph of a Sample Composition of Subsystems S_A and S_B

Figure 3.1 Subsystems S_A, S_B and Composite System S for Example 3.1

Lemma 3.1.2

The closure indices of the composing subsystems S_A, S_B and the composite system S are related by $c(S) \geq \min (c(S_A), c(S_B))$.

Proof:

If $c(S_A) = c(S_B) = \infty$ then assume that c(S) is finite. Let F^g be a closed set of faults in S. Let F^h be the set of faults such that $F^h = F^g \cap \mathfrak{F}_A$. Such a set must exist or else there exists a closed set of faults in S_B. Since $\mathfrak{F}_A$ is not closed, there must exist some test $t_i \in t_A(F^h)$ such that $t_i \notin T_A(F^h)$. By the definition of a composite system, $t_i \notin T(F^g)$ either, implying that F^g cannot be closed.

If either $c(S_A)$ or $c(S_B)$ is finite, then assume that $c(S) < \min (c(S_A), c(S_B))$. The argument is now similar to the one made in the preceding paragraph. Let F^g be a closed set of faults in S such that $|F^g| < \min (c(S_A), c(S_B))$. Let F^h be the set of faults such that $F^h = F^g \cap \mathfrak{F}^A$. Such a set must exist or else F^h would be a closed set of faults in S_B with cardinality less than $c(S_B)$. Since $|F^h| < c(S_A)$, F^h is not closed and there exists some test $t_i \in t_A(F^h)$ such that $t_i \notin T_A(F^h)$. By the definition of a composite system, $t_i \notin T(F^g)$ either, contradicting the assumption that F^g is closed. Q.E.D.

Theorem 3.1

The diagnosability with repair of a composite system is at least as large as the smaller of the diagnosabilities of its two composing systems, if the composing systems are STPF systems.

Proof:

By Lemma 3.1.2 $c(S) \geq \min (c(S_A), c(S_B))$. Let d_A and d_B be the diagnosabilities of S_A and S_B respectively. Then $c(S_A) > \left\lfloor \left(\frac{d_A+2}{2} \right)^2 \right\rfloor$ and

$c(S_B) > \left\lfloor \left(\frac{d_B+2}{2}\right)^2 \right\rfloor$. Let d be the diagnosability of the composite system. Since $c(S) > \min\left(\left\lfloor \left(\frac{d_A+2}{2}\right)^2 \right\rfloor, \left\lfloor \left(\frac{d_B+2}{2}\right)^2 \right\rfloor\right)$, it must be that $c(S) > \left\lfloor \left(\frac{\min(d_A,d_B)+2}{2}\right)^2 \right\rfloor$, d_A and d_B both being positive integers. A sufficient condition for the d-diagnosability of a system with repair is $c(S) > \left\lfloor \left(\frac{d+2}{2}\right)^2 \right\rfloor$ by Lemma 2.8.3. Hence $d \geq \min(d_A, d_B)$. Q.E.D.

Lemma 3.2.1

The diagnosability with repair of a system S whose closure index is c(S) is at least $\left\lfloor \sqrt{4c(S)-3} - 2 \right\rfloor$. In the special case when $c(S) = \infty$, the diagnosability with repair is n.

Proof:

Follows from Lemma 2.8.3 and some routine mathematical computation.

Lemma 3.2.2 [4]

A necessary condition for a system S to be d-fault diagnosable with repair is that the closure index for the system $c(S) \geq 2d + 1$.

Lemma 3.2.3

In a strongly connected self-testing SMPT system, c(S) = n, where n is the number of faults in the system.

Proof:

Let $F^g \in F$ be a closed set of faults in S. Assume that $|F^g| < n$. Then by the strongly connected nature of the diagnostic graph, there must exist a fault $f_i \in \mathfrak{F}$, such that $T(f_i) \cap t(F^g) \neq \emptyset$. Let $t_j \in t(f_j)$ be the test such that $t_j \in T(f_i) \cap t(F^g)$. By the SMPT assumption, there exists no other fault $f_k \in \mathfrak{F}$, $f_k \neq f_i$, such that $T(f_k) = t_j$. Hence $t_j \notin T(F^g)$ and $t(F^g) \not\subseteq T(F^g)$ implying F^g cannot be closed. Thus the only closed set of faults in the system is $\mathfrak{F}$, $|\mathfrak{F}| = n$, and c(S) = n. Q.E.D.

Theorem 3.2

If d_A and d_B are the diagnosabilities with repair of two composing subsystems S_A and S_B of a strongly connected self-testing SMPT composite system S, then the diagnosability with repair of S is at least $\lfloor \sqrt{8(d_A+d_B)+5} - 2 \rfloor$.

Proof:

If the composite system is SMPT, then the composing systems must also be SMPT. Since the diagnosabilities with repair of S_A and S_B are d_A and d_B, $|\mathfrak{F}_A| \geq 2d_A + 1$ and $|\mathfrak{F}_B| \geq 2d_B + 1$ by Lemma 3.2.2. Hence $|\mathfrak{F}| \geq 2(d_A + d_B + 1)$. Hence $c(S) = |\mathfrak{F}| \geq 2(d_A + d_B + 1)$ by Lemma 3.2.3. From Lemma 3.2.1, it then follows that the diagnosability with repair of the system is at least $\lfloor \sqrt{4.2(d_A+d_B+1)-3} - 2 \rfloor$ or $\lfloor \sqrt{8(d_A+d_B)+5} - 2 \rfloor$. Q.E.D.

It must be noted here that the knowledge of c(S) for the composite system will give a better lower bound for the diagnosability. The above theorem is useful when nothing is known about the composing subsystems except their diagnosabilities.

Theorem 3.3

If d_A and d_B are the maximum diagnosabilities with repair of two composing subsystems S_A and S_B of a strongly connected self-testing SMPT composite system S, then the diagnosability with repair of S is at most $\lfloor (d_A+3)^2/8 + (d_B+3)^2/8 - 1 \rfloor$.

Proof:

Since the diagnosabilities with repair of S_A and S_B are d_A and d_B, $|\mathfrak{F}_A| \leq \left\lfloor \left(\frac{(d_A+1)+2}{2} \right)^2 \right\rfloor$ and $|\mathfrak{F}_B| \leq \left\lfloor \left(\frac{(d_B+1)+2}{2} \right)^2 \right\rfloor$. Hence $|\mathfrak{F}| \leq \lfloor (d_A+3)^2/4 + 1 \rfloor + \lfloor (d_B+3)^2/4 + 1 \rfloor$. Since d_A is an integer, $(d_A+3)^2/4 - \lfloor (d_A+3)^2/4 \rfloor$ is either 0 or $\frac{1}{4}$. Hence it is valid to say that

$|\mathcal{F}| \leq \lfloor (d_A+3)^2/4 + (d_B+3)^2/4 - 1 \rfloor$. From Lemma 3.2.2 it then follows that the diagnosability with repair of S is at most $\frac{1}{2} \lfloor (d_A+3)^2/4 + (d_B+3)^2/4 - 2 \rfloor$ or $\lfloor (d_A+3)^2/8 + (d_B+3)^2/8 - 1 \rfloor$. Q.E.D.

Example 3.2

Consider the composite system shown in Figure 3.2 which is composed of two single-loop subsystems $\mathcal{F}_A = \{f_1, f_2, f_3\}$ and $\mathcal{F}_B = \{f_4, f_5, f_6, f_7, f_8\}$. $\mathcal{T} - \mathcal{T}_A - \mathcal{T}_B = \{t_9, t_{10}\}$.

It can be easily shown [4] that $d_A = 1$ and $d_B = 2$. By Theorem 3.2, $d \geq \lfloor \sqrt{8(d_A+d_B)+5} - 2 \rfloor$. Hence $d \geq \lfloor \sqrt{8*3+5} - 2 \rfloor = \lfloor 5.38 - 2 \rfloor = 3$. By Theorem 3.3, $d \leq \lfloor (d_A+3)^2/8 + (d_B+3)^2/8 - 1 \rfloor$. Hence $d \leq \lfloor 16/8 + 25/8 - 1 \rfloor = 4$.

(It may be verified that the actual diagnosability with repair of the composite system is 3.) □

Theorems 3.2 and 3.3 are useful in determining the upper and lower bounds on the diagnosability of a system constructed from two systems of known diagnosabilities. It must be mentioned that the requirement that the composite system be SMPT, strongly connected and self-testing is somewhat restrictive; however, most of the systems that have been considered in the literature, like the single loop systems and the $D_{\delta A}$ systems [1,18] fall in this category.

3.3. General Cases

For the more general cases, direct results like those obtained for the SMPT case are more difficult to obtain. Usually, the closure index for a subsystem would be smaller than the number of faults in the single fault set. In such cases, the extent to which the diagnosability of the composite system is greater than that of the composing subsystems depends on how the subsystems are connected.

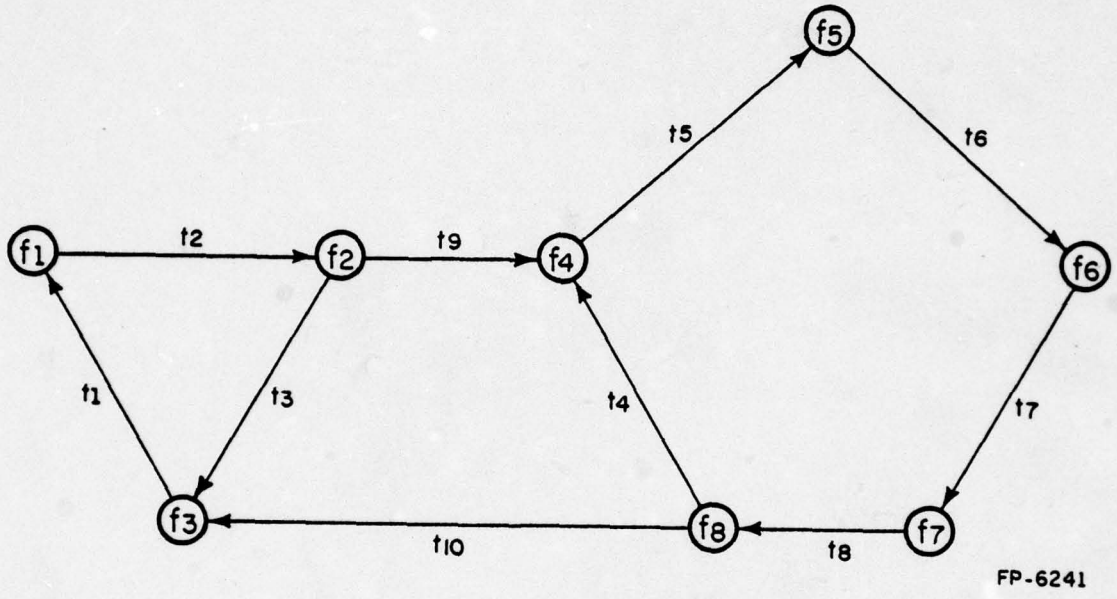


Figure 3.2 Composite SMPT System for Example 3.2

Lemma 3.4 [4]

A system $S = (\mathcal{F},\mathcal{T},F,G)$ is d-fault diagnosable with repair if and only if, for any $F^i,F^j,\ldots,F^h \in F(d)$, $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ implies $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$.

A more general version of Theorem 3.1 may now be stated.

Theorem 3.4

The diagnosability with repair of a composite system is at least as large as the smaller of the diagnosabilities of its two composing subsystems.

Proof:

Let the system be $S = (\mathcal{F},\mathcal{T},F,G)$ and its two composing subsystems be $S_A = (\mathcal{F}_A,\mathcal{T}_A,F_A,G_A)$ and $S_B = (\mathcal{F}_B,\mathcal{T}_B,F_B,G_B)$. Let d, d_A and d_B be the diagnosabilities of S, S_A and S_B. Without loss of generality it could be assumed that $d_A \leq d_B$. Consider a set offaults, $\{F^m,F^n,\ldots,F^r\} \subseteq F(d_A)$ such that $F^m \cap F^n \cap \ldots \cap F^r = \emptyset$. Hence $F^m \cap F^n \cap \ldots \cap F^r \cap \mathcal{F}_A = \emptyset$. Let $F^i = F^m \cap \mathcal{F}_A$, $F^j = F^n \cap \mathcal{F}_A$, ..., $F^k = F^r \cap \mathcal{F}_A$. Clearly, $\{F^i,F^j,\ldots,F^k\} \subseteq F_A(d_A)$ and $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$. By Lemma 3.4, $G_A^i \sqcap G_A^j \sqcap \ldots \sqcap G_A^k = \emptyset$. By the definition of a composite system, no fault in $\mathcal{F}_B$ can mask a test $t \in \mathcal{T}_A$. Hence $G^m \sqcap G^n \sqcap \ldots \sqcap G^r = \emptyset$. Since $\{F^m,F^n,\ldots,F^r\} \subseteq F(d_A)$ and $F^m \cap F^n \cap \ldots \cap F^r = \emptyset$ implies $G^m \sqcap G^n \sqcap \ldots \sqcap G^r = \emptyset$, S must be d_A-fault diagnosable with repair. Q.E.D.

It is not difficult to find examples of systems which have diagnosabilities as small as the diagnosability of one of the composing subsystems. The case when $\mathcal{T} = \mathcal{T}_A \cup \mathcal{T}_B$ provides one such example.

Similarly, given a system, one may be interested in augmenting the diagnosability of the system by adding additional tests and testing

units. This would imply the augmentation of the diagnostic graph for the system, with additional fault nodes and links representing tests. From the above reasoning it may be clear that certain types of augmentation of the system diagnostic graph may not improve the diagnosability of the system at all. Conversely there may be certain "critical" faults in the system, adding tests for which could help in improving the system diagnosability.

The j-open circuit diagnosability of a system S, denoted by $d_j^o(S)$ is defined as the diagnosability of the system S', which differs from S in that $\mathcal{T}' = \mathcal{T} \cup t_h$, where t_h is a new complete test for f_j and t_h is not invalidated by any other fault in the system.

The similarity of this measure with the open-circuit condition in conventional electrical networks will be discussed later. The significance of this measure will now be demonstrated.

A subsystem $S_1 = (\mathfrak{F}_1, \mathcal{T}_1, F_1, G_1)$ is said to be j-in-connected to a system $S = (\mathfrak{F}, \mathcal{T}, F, G)$ if in the composite system $S' = (\mathfrak{F}', \mathcal{T}', F', G')$ there exist some faults f_i and f_j such that $T(f_i) \cap t(f_j) \neq \emptyset$ (i.e., f_i invalidates some test for f_j) and $f_i \in F_1$, $f_j \in F$. Also, in this case S_1 would be said to be i-out-connected to system S. In the diagnostic graph for the composite system this would be indicated by an arc emanating from node i and directed into node j.

Example 3.3

Consider the system whose diagnostic graph is shown in Figure 3.3. The diagnosability with repair of the system is 3. The 6-open-circuit diagnosability of the system is 12, while the 10-open-circuit diagnosability of the system remains 3.

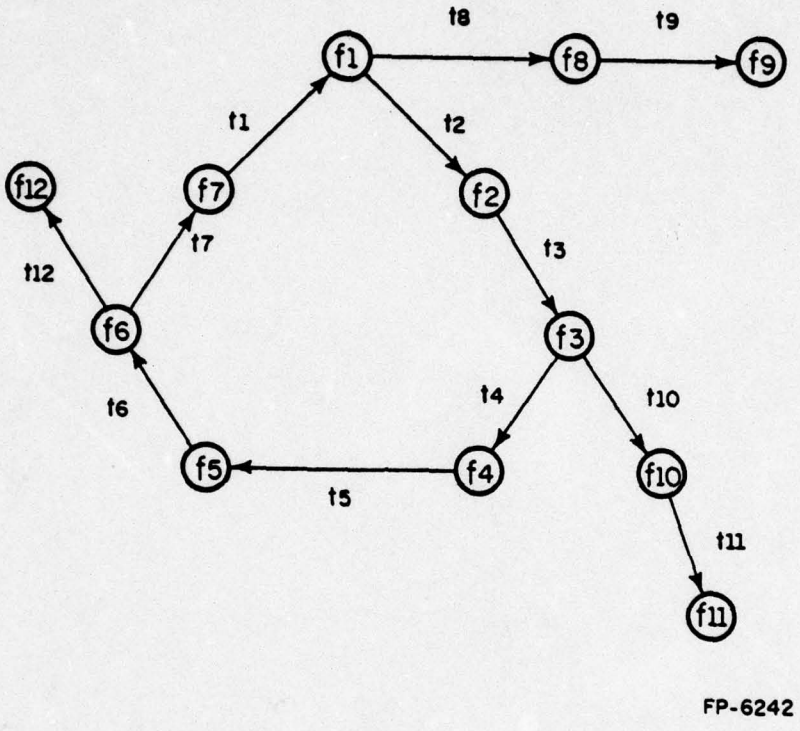


Figure 3.3 Diagnostic Graph for System in Example 3.3

Consider the syndrome 100010000000. If 4 faults were allowed in the system, then both $\{f_1,f_2,f_3,f_4\}$ and $\{f_5,f_6,f_7\}$ could have yielded the same syndrome. Hence the system is not 4-fault diagnosable with repair. Since $c(S) = 7$, the system is 3-fault diagnosable with repair. When another test for f_6 is added, and this new test cannot be invalidated, then the structure of the graph ensures that the presence of at least one fault can be detected if any fault occurs. On the other hand, if a new test is added for f_{10}, a fault can be detected only if at most 3 faults occur or if the faults which occur include f_{10} or f_{11}. □

Theorem 3.5

The diagnosability d of a system S can be improved by j-in-connecting another subsystem to S only if $d_j^o(S) > d$.

Proof:

Consider the composite system S' obtained by j-in-connecting S_1 to S. Let d', the diagnosability of S' be greater than d. Since $d' > d$ and $d_j^o(S') \geq d'$, $d_j^o(S') > d$. Consider the system $S_2 = \{\mathfrak{F}_2,\mathcal{T}_2,F_2,G_2\}$ where $\mathfrak{F}_2 = \mathfrak{F}_1$, $\mathcal{T}_2 = \mathcal{T}_1 \cup t_h$ where $t_h \in t(f_j)$ and $t_h \notin T(\mathfrak{F})$. Thus t_h is a test for f_j that is not invalidated by any other fault in the system. Let $S_3 = \{\mathfrak{F}_3,\mathcal{T}_3,F_3,G_3\}$ be another system where $\mathfrak{F}_3 = \mathfrak{F}'$, $\mathcal{T}_3 = \mathcal{T}' \cup t_h$. It must be the case that $d_2 = d_j^o(S') > d$ and $d_3 = d_j^o(S) \geq d$. Consider the set of faults $\{F^g,F^h,\ldots,F^m\} \subseteq F_3(d+1)$. Since $\mathfrak{F}_3 \subseteq \mathfrak{F}_2$, $\{F^g,F^h,\ldots,F^m\} \subseteq F_2(d+1)$, and since S_2 is at least $(d+1)$ diagnosable without repair, $F^g \cap F^h \cap \ldots \cap F^m = \emptyset$ implies $G^g \sqcap G^h \sqcap \ldots \sqcap G^m = \emptyset$. Since $\{F^g,F^h,\ldots,F^m\} \subseteq F_3(d+1)$ and $F^g \cap F^h \cap \ldots \cap F^m = \emptyset$ implies $G^g \sqcap G^h \sqcap \ldots \sqcap G^m = \emptyset$, S_3 is $(d+1)$ diagnosable with repair by Lemma 3.4. Hence $d_j^o(S) \geq d + 1$. Q.E.D.

The condition mentioned above is not sufficient. This can be illustrated by the example of a STPF composite system in which the closure index does not change due to the in-connection of one subsystem to another.

Example 3.4

Figure 3.4 shows a system taken from [4] which is 2-fault diagnosable with repair.

If another system is to be in-connected to the system shown at A1, then no improvement in diagnosability can result, because the closure index for the composite system can be no greater than 5. (A closed set of faults is {R1,C1,C2,R2,M1}.) On the other hand, analysis shows that $d^{o}_{C1}(S)$, the open circuit diagnosability for fault C1, is 9, implying that an improvement in diagnosability could result by in-connecting another system to C1. □

Theorem 3.5 essentially indicates the nodes in the diagnostic graph which are critical in the sense described earlier. Thus, there may exist many faults in the system, adding tests for which may not help in augmenting the diagnosability of the graph. In some systems, like STPF systems, the closed set of faults which define the closure for the system is the "critical" set.

Depending on the closure indices of the connecting subsystem and the connected subsystems, the diagnosability of the composite system may or may not be greater than the diagnosability of any of the subsystems even if the conditions of Theorem 3.5 were satisfied. An intuitive idea of this may be obtained from Lemma 3.1.3 from which we may observe that if $c(d,S)$ is the required closure index to ensure d-diagnosability with repair for a system S, then $c(d+1,S) - c(d,S)$ increases as d increases. (This increase is, in fact, linear).

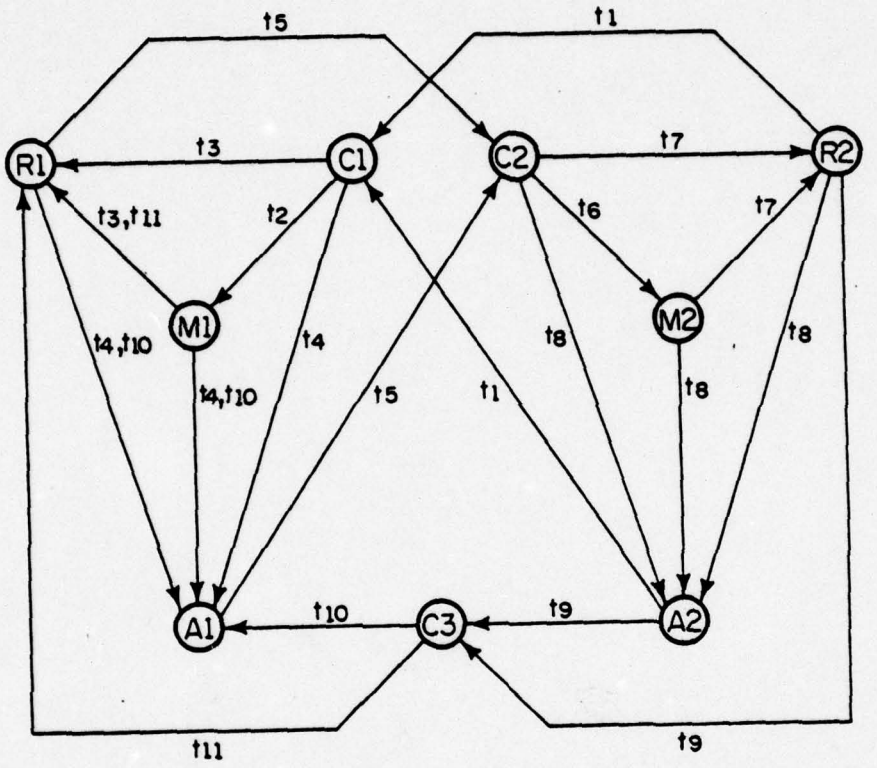


Figure 3.4 Diagnostic Graph for System in Example 3.4

Generally, when two subsystems are connected together to form a composite system, the interconnections between the two subsystems may take various forms. The interconnection will be termed <u>directional</u> ($S_A \rightarrow S_B$), if in the diagnostic graph of the composite system, all the edges between the two subsystems have their origin in one subsystem (S_A) and are directed towards the other subsystem (S_B). The interconnection will be termed <u>non-directional</u> otherwise. Two subsystems are called <u>n-connected</u> if there are n edges between the two subsystems. Thus two subsystems which are non-directionally 2-connected will have exactly one edge emanating from each of the subsystems. This implies that a test for some fault in each of the subsystems is masked by some fault in the other subsystem.

<u>Theorem 3.6</u>

If two strongly connected SMPT subsystems S_A and S_B are directionally interconnected, $S_A \rightarrow S_B$, then the diagnosability of the composite system must be at least as large as the diagnosability of S_A.

<u>Proof</u>:

(We assume that the composite system remains SMPT for tests for all faults in $\mathfrak{F}_B$.) Let $\{F^i,F^j,\ldots,F^k\} \subseteq F(d)$ be a set of faults in the composite system where d is the diagnosability of S_A. If $\{F^i,F^j,\ldots,F^k\} \subseteq F_A$ then, by assumption, $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ implies $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$. If $\{F^i \cup F^j \cup \ldots \cup F^k\} \cap F_A \neq \emptyset$ then $\{F^i \cap F_A\} \cap \{F^j \cap F_A\} \cap \ldots \cap \{F^k \cap F_A\} = \emptyset$ implies $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$ because none of the tests for faults in $\mathfrak{F}_A$ are invalidated by some fault in $\mathfrak{F}_B$. If, on the other hand, $\{F^i \cup F^j \cup \ldots \cup F^k\} \cap F_A = \emptyset$ and $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$, then there exists some fault $f_p \in F^g$, $f_p \notin F^h$, where $F^g,F^h \in \{F^i,F^j,\ldots,F^k\}$ such that $t(f_p) \notin T(F^i \cup F^j \cup \ldots \cup F^k)$. This is true because otherwise $F^i \cup F^j \cup \ldots \cup F^k$

would be a closed set of faults with cardinality less than $|\mathfrak{F}_B|$ which contradicts the strongly connected SMPT assumption for the subsystem. If $t(f_p) = t_p$ then clearly $G_p^g \sqcap G_p^h = \emptyset$ and hence the composite system must remain at least d-diagnosable with repair. Q.E.D.

An interesting facet of Theorem 3.6 is that the result does not depend on the diagnosability of S_B, as long as S_B is strongly connected and SMPT. In fact, it may also be seen that it is not necessary for S_A to be strongly connected or SMPT. The following corollaries to Theorem 3.6 can be easily proved.

Corollary 3.6.1: If a subsystem S_A with diagnosability d is directionally interconnected, $S_A \rightarrow S_B$, to a strongly connected SMPT subsystem S_B, the diagnosability of the composite system is at least as large as d_A, the diagnosability of S_A.

Corollary 3.6.2: Let d_A and d_B be the diagnosabilities of the two composing subsystems of a strongly connected SMPT composite system with non-directional interconnection. The diagnosability d of the composite system obeys the relation $d \geq \max(d_A, d_B)$.

The above result is useful in the sense that it gives an assurance that the diagnosability in a SMPT system can never deteriorate by interconnecting with other SMPT systems of any size. In fact, the proof of Theorem 3.6 indicates that there are other systems for which similar results may be obtained.

Theorem 3.7

If two subsystems are directionally interconnected, $S_A \rightarrow S_B$, and the closure index of S_B is $|\mathfrak{F}_B|$, then the diagnosability of the composite system must be at least as large as the diagnosability of S_A.

Proof:

Almost identical to that of Theorem 3.6.

Example 3.5

Let S_A be a single-loop system consisting of 10 nodes and S_B be an arbitrary 5 node system with closure index of 5. Let the directional interconnection $S_A \rightarrow S_B$ be as shown in Figure 3.5.

The diagnosability of S_A is 4. The closure index for the composite system may be verified to be 15. Hence the diagnosability of the composite system is at least $\lfloor \sqrt{4 \times 15 - 3} - 2 \rfloor$ or 5 by Lemma 3.2.1. This agrees with Theorem 3.7 by which the diagnosability must be at least 4. □

A composition of two subsystems in which the interconnection between the subsystems consists of new complete tests which were not originally tests for either of the subsystems will be called a constructive composition. Such an interconnection will be termed a constructive interconnection. Thus, if $S_A = (\mathfrak{I}_A, \mathcal{T}_A, F_A, G_A)$ and $S_B = (\mathfrak{I}_B, \mathcal{T}_B, F_B, G_B)$ are the two composing subsystems for a composite system $S = (\mathfrak{I}, \mathcal{T}, F, G)$ then $T(\mathfrak{I}_A) \cap T_B(\mathfrak{I}_B) = \emptyset$ and similarly $T(\mathfrak{I}_B) \cap T_A(\mathfrak{I}_A) = \emptyset$.

Corollary 3.7.1: Two subsystems S_A and S_B with closure indices $|\mathfrak{I}_A|$ and $|\mathfrak{I}_B|$ connected with a non-directional, constructive interconnection yield a composite system with diagnosability d, where $d \geq \max(d_A, d_B)$.

Proof:

Without loss of generality it may be assumed that $d_A \geq d_B$. Let $\{F^i, F^j, \ldots, F^k\} \subseteq F(d_A)$, but $\{F^i, F^j, \ldots, F^k\} \not\subseteq F(d_B)$. As in the proof of Theorem 3.6, if $\{F^i \cup F^j \cup \ldots \cup F^k\} \cap F_A \neq \emptyset$ then $\{F^i \cap F_A\} \cap \{F^j \cap F_A\} \cap \ldots \cap \{F^k \cap F_A\} = \emptyset$ implies $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$ because none of the tests for faults in $\mathfrak{I}_A$ is invalidated by any fault in $\mathfrak{I}_B$. If $\{F^i \cup F^j \cup \ldots \cup F^k\} \cap F_A = \emptyset$,

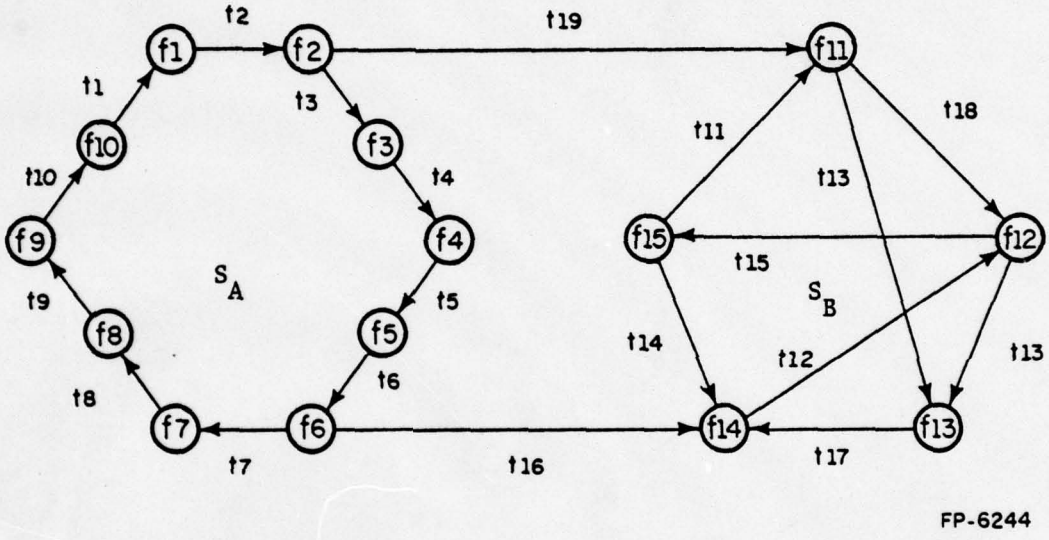


Figure 3.5 Directional Interconnection in the Composite Graph of Example 3.5

then the assumption that $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ implies the existence of some test t_p, $t_p \in t(F^g)$, $t_p \notin t(F^h)$, where $F^g, F^h \in \{F^i, F^j, \ldots, F^k\}$ such that $t_p \notin T(F^i \cup F^j \cup \ldots \cup F^k)$. This is true or else $F^i \cup F^j \cup \ldots \cup F^k$ would form a closed set of faults with cardinality less than $|\mathfrak{F}_B|$ contrary to assumption. Hence $G_p^g \sqcap G_p^h = \emptyset$. Thus $F^i \cap F^j \cap \ldots \cap F^k = \emptyset$ implies $G^i \sqcap G^j \sqcap \ldots \sqcap G^k = \emptyset$ and the system must be d_A-diagnosable with repair. Q.E.D.

Corollary 3.7.2: When two subsystems S_A and S_B with closure indices $|\mathfrak{F}_A|$ and $|\mathfrak{F}_B|$ are interconnected non-directionally, the composite system has a diagnosability $d < \max(d_A, d_B)$ only if the interconnection is non-constructive.

Proof:

Direct consequence of Corollary 3.7.1.

The importance of these corollaries lies in the fact that they indicate the desirable ways to interconnect subsystems without decreasing the diagnosability.

3.4. Discussion

This chapter has attempted to determine the behavior of system diagnosability when the system is augmented. When the systems that are being combined are of the single-mask-per-test (SMPT) type some convenient upper and lower bounds for the diagnosability of the composite system may be obtained. However, for the general cases, the results are not as convenient.

The open-circuit diagnosability of the system is a good way to determine if a certain interconnection with another system can improve the diagnosability of the system. An analogy can be made between this measure and the open-circuit voltage in an electrical network.

For instance, the open-circuit voltage between terminals a and b of the electrical circuit of Figure 3.6 gives an indication of the current that will pass through a passive resistor R connected across those terminals. If the open-circuit voltage is zero, then no current will pass through R. However, if the open-circuit voltage is non-zero then the amount of current that passes through R is not immediately known; the resistance of R and the Thevenin's equivalent resistance of the network itself must be known, in addition.

Similarly, the open-circuit diagnosability of a computer network indicates whether the increase in diagnosability of a system when another subsystem is connected to it is zero or non-zero. As in the case of electrical networks, the actual increase in the diagnosability depends not only on the j-open-circuit diagnosability but also on other aspects of the original system and the interconnected subsystem. Further, if the interconnected subsystem is trivially a global mechanism which can test for fault j without invalidation, then the actual diagnosability of the system equals the j-open-circuit diagnosability.

An added complication in the determination of the diagnosability of composite systems is the directionality of interconnections. A few results have been obtained for both directionally and non-directionally interconnected subsystems. It may be interesting to study systems in which, if a test for fault f_i is invalidated by fault f_j, then some test for f_j must also be invalidated by f_i. The complete duality of the directionality of interconnections in such cases may make them more amenable to analysis.

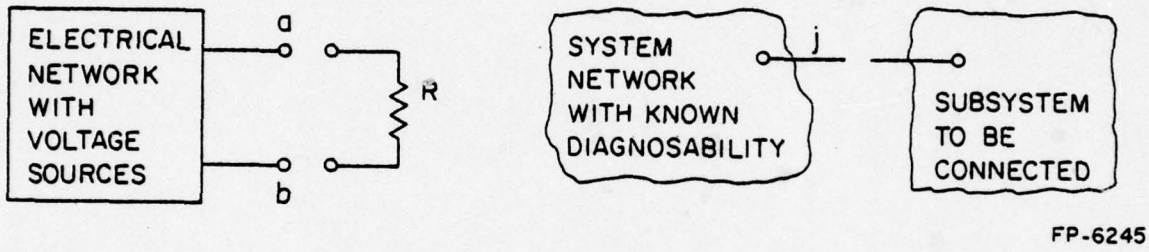


Figure 3.6 Significance of Open-Circuit Measures

Two benefits may be cited for studies of composite system diagnosability as in this chapter. First, they aid in determining desirable and undesirable interconnections (from the diagnosability point of view) to augment existing systems. Thus it is quite possible that the diagnosability of a system actually deteriorates when more nodes are connected to the system, if sufficient care is not taken with the diagnostic procedures. Second, once the interconnections have been made in the appropriate manner, the system still may be viewed as a composition of subsystems from the point of view of diagnosis. This could facilitate testing because by considering smaller networks at a time the diagnostic routines, which are at best NP-complete in computational complexity, will be able to operate efficiently and economically. Since the global mechanism to analyze test syndromes, whether implemented in hardware or in software, is assumed to be error-free, the reduced complexity implied by smaller subsystems could be of advantage.

4. SELF-DIAGNOSIS STRATEGIES FOR HIGH PERFORMANCE DISTRIBUTED SYSTEMS

4.1. Introduction

Chapter 1 gave an overview of the techniques that have been used in the past in system diagnosis. Most of these were used to analyze the diagnosability of systems whose diagnostic graphs were known. Very often, such research addresses the problem of optimal designs rather than designs for systems that are easily implementable and easily diagnosable, and consequently the problem tends to become formidable even for the simplest of configurations.

Simplicity of design, rather than optimality, is gaining in importance as component costs continue to decrease. The time saved by using a near-minimal realization instead of trying to obtain a minimal realization frequently offsets the cost of extra components.

Another observation is the implicit assumption of some global mechanism, call it a global supervisor, which is capable of collecting data and of sending commands to the various subsystems. Thus in the connection assignment problem [1], this global supervisor can collect the results of the tests performed by the various units and then decide in a "one-step" or "sequential" manner which units are faulty. In the Hayes approach [10,21], the supervisor can test the units and can reconfigure the units appropriately in the event of a fault. In the test-point and blocking gate approach [13,14], the global mechanism has control over the stimuli that have to be propagated through the system and can observe the response of the system to these stimuli.

In all of the above instances and in the techniques of the previous chapters of this thesis, the global supervisor assumes the major role of ensuring proper functioning of the system while remaining fault-free itself. Such an assumption, while not unreasonable for ground-based systems, certainly proves to be a drawback in space- and air-borne systems. As the functions it is required to perform increase in complexity, the reliability of the global supervisor itself assumes greater importance.

The need is hence felt for a diagnosis strategy wherein the global hard-core requirements in the system are minimized, if not eliminated, so that the system is capable of testing, diagnosing and reconfiguring itself with little or no external help. The rest of this thesis presents a few ideas in this direction.

4.2. System Model

The system being considered will be assumed to be composed of many smaller subsystems (or units or modules) interconnected physically to form a network. Each subsystem may be a complex processing facility with its own memory like a microcomputer or even a large computer (as in a distributed nationwide network) or else it may be a small digital device which is incapable of standing alone, like an I/O device which is not intelligent. The only assumption being made about the subsystem is that by using only its input and output ports, each subsystem can be tested completely for faults in a reasonable fault model. This is a major assumption, particularly because satisfactory solutions to the test generation problem in microcomputers and other large-scale integrated circuits have yet to appear in the literature.

It is convenient to separate the model which represents the communication capability between units in the network from the model which indicates the testing capabilities of the various units. In order to retain consistency with the earlier chapters, the graph for the testing model will be identical to the diagnostic graph of Russell and Kime [4]. A communications graph could also be defined for the system where each node corresponds to a subsystem or unit and an edge from node v_i to node v_j indicates that messages/data can be sent from the facility in the system corresponding to v_i to that corresponding to v_j. Further, if a one-to-one correspondence exists from the set of faults in the diagnostic graph to the set of units in the communications graph, the terms fault and faulty unit may be used interchangeably.

While the communications graph could conceivably be a directed graph it will often be an undirected graph. This simply means that if communication is possible between two nodes in a system, it is generally possible in both directions. Further, in such cases, the diagnostic graph would form a subset of the communications graph, because a faulty node f_i is unlikely to affect the results of a test for a faulty node f_j if the two nodes are not physically connected. It must be cautioned, however, that a directed edge from f_i to f_j in the diagnostic graph need not always imply a directed edge from the node corresponding to f_i to that corresponding to f_j in the communications graph. For instance, when one unit corresponding to f_i simply receives the results of a test from another corresponding to f_j, the communications graph need have only a link from the latter node to the former, while the diagnostic graph will possess a directed arc from f_i to f_j.

4.3. A Strategy for Diagnosis

This thesis aims at the design of systems capable of continuous self-diagnosis by using the principle of "roving diagnosis." In essence, roving diagnosis requires one part of the system to be diagnosing another, while the remainder of the system continues normal operation. The part of the system most recently diagnosed as fault-free now takes its turn to diagnose another part. Thus there is a subsystem involving the diagnosing and diagnosed units which apparently roves through the system until no part of the system remains undiagnosed. This is in contrast to non-roving schemes, where either

(i) the system is diagnosed as a whole, by an external tester, or

(ii) the system is divided into modules which may be independently tested, but still employs an external tester (the "roving doctor" scheme), or

(iii) the system is divided into identical modules performing identical tasks, with a vote on their results determining their validity, or

(iv) the system is divided into two halves, each diagnosing the other.

The problem with (i) and (ii) above is that the reliability of the external tester (or "the doctor") is never questioned. In (iii), the system is never utilized anywhere close to its potential, although its reliability can be made quite high for an arbitrary but limited time. (iv) suffers from the disadvantage of having to shut down a major portion of the system during diagnosis, without even ensuring reliable results.

The roving diagnosis approach is an attempt to solve one of the problems of Section 4.1, namely that of eliminating the global supervisor

in systems. It also helps to keep from shutting the entire system down for diagnosis and involves as little a portion of the system as possible for diagnosis. The extent to which the approach is capable of doing away with the global supervisor depends a great deal on the assumed fault model for the system. Thus there exist faults whose identification is beyond the capability of even a global arbiter.

There are a number of ways in which system testing may actually be performed for roving diagnosis:

a) Testing is performed throughout the system at preset intervals of time. Once testing begins, all units in the system are tested with as little a time-gap between tests in successive areas as possible. To maintain high system utilization, areas not involved in diagnosis continue computation (or processing).

b) The intervals of time at which a particular unit is tested remain constant and the same as in (a). However, the testing of successive areas need not be immediate.

The major difference between the two approaches is that in (a) there are fewer, but longer, intervals of time when the entire system is involved in computation only. In (b) the periods of "computation only" are almost equally interspersed with the periods when some area is involved in diagnosis. Figure 4.1 attempts to illustrate this.

The actual time taken for testing any unit will depend on

(i) length of test routine (which also depends on the complexity of the unit and the fault model),

(ii) the number of units involved in diagnosis, and

(iii) the speed of the diagnosing and diagnosed units.

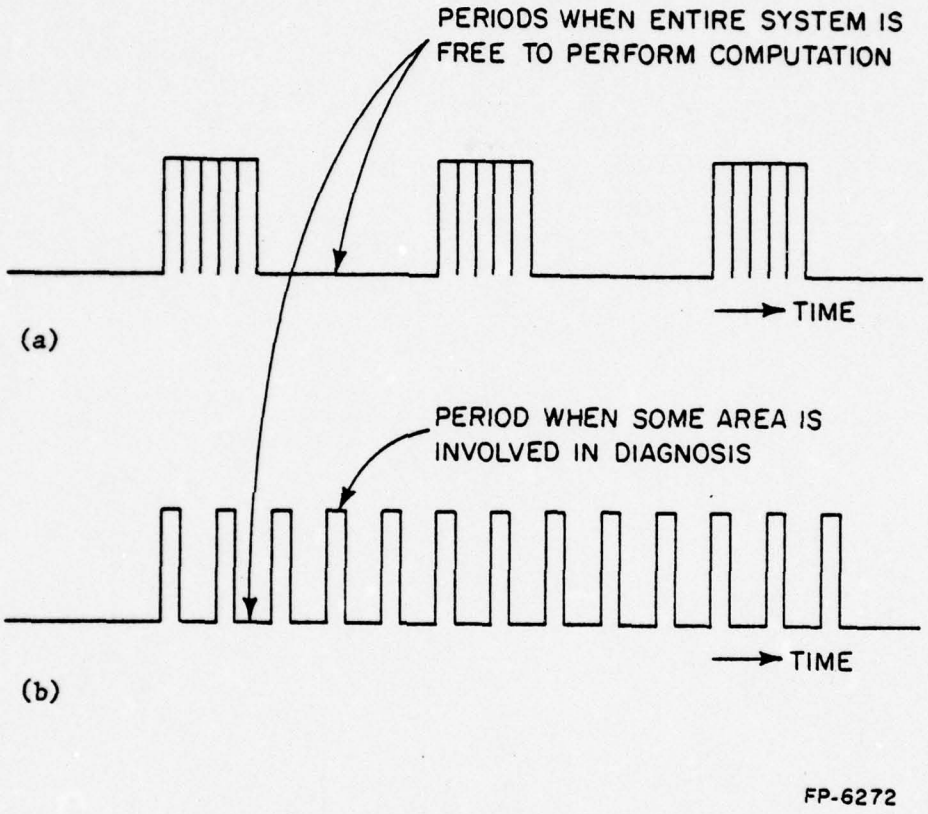


Figure 4.1 Comparison of Approaches for Testing

The time between tests for any unit will depend on

(i) the mean time between failures for that unit, and

(ii) whether or not the unit is self-checking. (If the unit is self-checking, then another unit could be continuously monitoring the output of its checker. Thus failures during operation could be caught fairly easily. However, since the exercising of all possible code inputs during regular operation is unlikely, a routine testing of the unit is necessitated, though possibly at reduced intervals of time.)

4.4. System Reconfiguration

In any self-diagnosing strategy, roving diagnosis being no exception, the occurrence and the identification of a fault must be accompanied by the broadcasting of this fact to the concerned units in the area of the fault, if not to the entire system. The region of dissemination of fault information is governed by the usefulness of the information to the system modules for purposes like reconfiguration.

Reconfiguration involves the logical disconnection of the faulty components before the system effects a recovery by restoring the integrity of the data and control to continue execution. If sufficient redundancy has been incorporated in the system, the performance of the system can continue at the same level as before without degradation in the average computing power. (The cost, of course, is reduced tolerance to further faults.) The discussion of the future chapters will deal more with whether computing is at all possible after reconfiguration and the tolerance to faults of the reconfigured system, rather than the degree of degradation of performance.

The reconfigurability of systems has been defined by different authors in different ways. For example, in the Hayes model [10], the system lends itself to reconfiguration on the occurrence of a fault only if the remaining system after removal of faulty units contains a subgraph isomorphic to the algorithm facility graph. For further discussion in this thesis, the following definitions are proposed:

A system will be deemed reconfigurable if all the neighbors can be reliably informed about the fault. (A neighbor of a unit is one which has a physical link to that unit.) Thus, on the occurrence of a fault, a system which is reconfigurable from that fault can sever all links to the faulty unit and attempt to continue functioning without that unit. This is important because the retention of faulty units in the system may result in misleading information, generated by the faulty unit, to be floating around the system.

A system will be termed reusable after a fault if the system is reconfigurable from the fault, and the communications graph of the reconfigured system has the minimum number of nodes and minimum links between the nodes necessary for the useful functioning of the system.

An example to illustrate these points is shown in Figure 4.2, which shows the communication graph for a system consisting of two types of units. Proper functioning of the system requires a connected communications graph consisting of at least two units of type 1 and one of type 2.

There exist at least two distinct paths between any pair of nodes in the graph. This ensures that any unit which finds its neighbor faulty can inform all the other neighbors of the faulty unit about the presence of the fault. This is no longer true if two or more faults can occur simultaneously

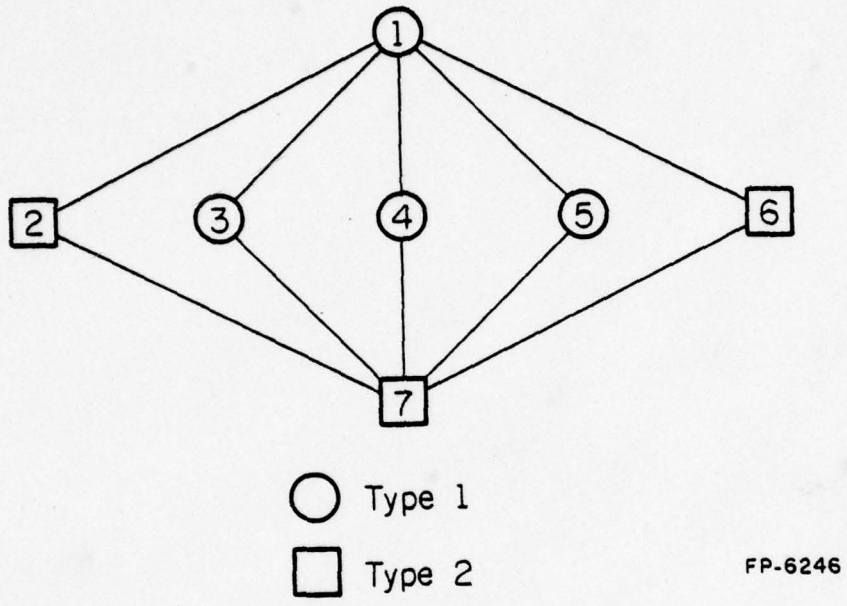


Figure 4.2 Communications Graph for a Sample System

in the system. Thus the system of Figure 4.2 is reconfigurable from at most one fault. In fact, the system remains connected after the occurrence of any single fault, with at least three nodes of type 1 and two of type 2. Hence the system is also reusable after a single fault. (By adding more links to the system, as shown in Figure 4.3, the system can be made reconfigurable and reusable after two faults.)

An example of a system which is reconfigurable from three faults but not reusable is shown in Figure 4.4. If any three units of a single type fail, the remaining system, though connected, will have fewer units of that type than required in a basic system.

4.5. Fault Assumptions

In order to eliminate any confusion, a fault will be considered to be synonymous with a faulty unit. This is not only convenient, but also is in keeping with the primary concern of locating faults down to a single unit or a single node in the communications graph. Thus, a single fault refers to a single faulty node. It may well be that a fault in this node involves many physically distinct faults, possibly of the stuck-type.

It will be assumed also that faults are restricted to the nodes in the communications graph. This assumption is justified for the following reasons:

(i) The complexity of the units being tested, generally of the microprocessor type, is much larger than that of the interconnections between the units, thereby making the former far more prone to faults.

(ii) Most communication between units, especially in large computer networks, is of the asynchronous, handshaking type. Hence

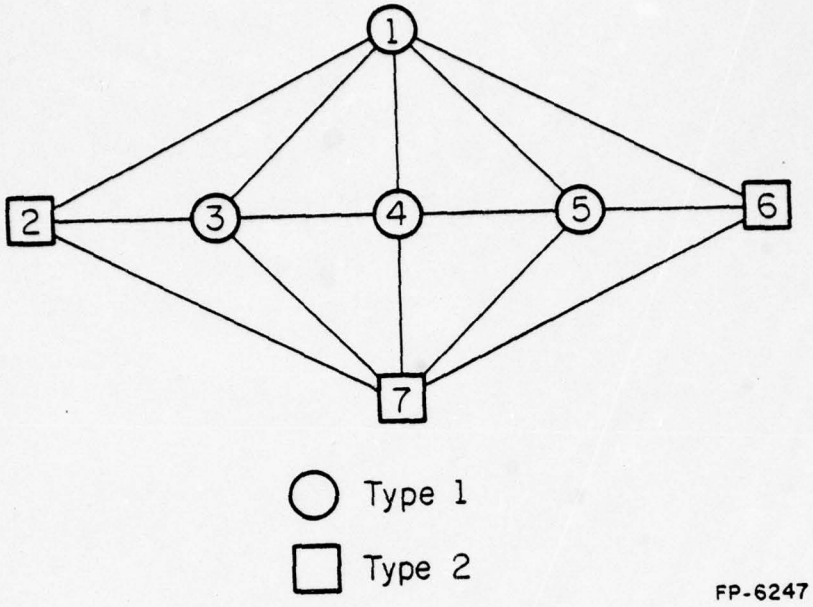


Figure 4.3 A 2-Fault Reconfigurable, Reusable System
(Basic System: 2 of type 1; 1 of type 2)

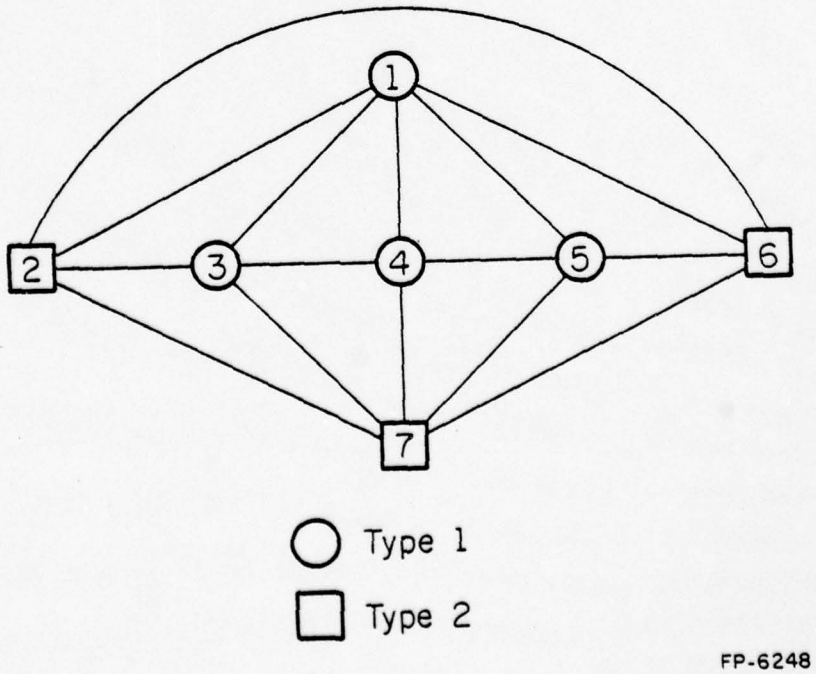


Figure 4.4 A 3-Fault Reconfigurable, Non-Reusable System
(Basic System: 2 of type 1; 1 of type 2)

obvious link faults can be detected during regular operation. (The receipt of an unintelligible message or a non-code input is an example.)

(iii) Parity bits or other error-detecting mechanisms may be used to further improve the reliability of links by making them diagnosable.

(iv) Very often, the faults on links dominate some faults on nodes, so that the testing of links in these cases is accomplished automatically during the testing of nodes.

(v) Schemes, like those used at present in computer networks, e.g., ARPANET [22], may be used to verify the liveness of links. At periodic intervals of time, messages are sent to and fro on every link, between units on the ends of the link. These messages will be essentially of the type, "Hello, I am fine; how are you?" and attempt to cover the possibility of links remaining unused for long periods of time. While such a scheme is not necessary to determine whether the link is up (the next handshake will definitely verify that), it helps in the early detection of failures and reduces the delay in informing the system about the fault.

Even if it is deemed necessary to incorporate the possibility of link failures into the system model, the links may be viewed as separate "facilities" and the system communications graph can incorporate nodes corresponding to the links. This was the technique used by Hayes [10] in his facility graph approach. Figure 4.5 shows a sample conversion to incorporate the possibility of link failures.

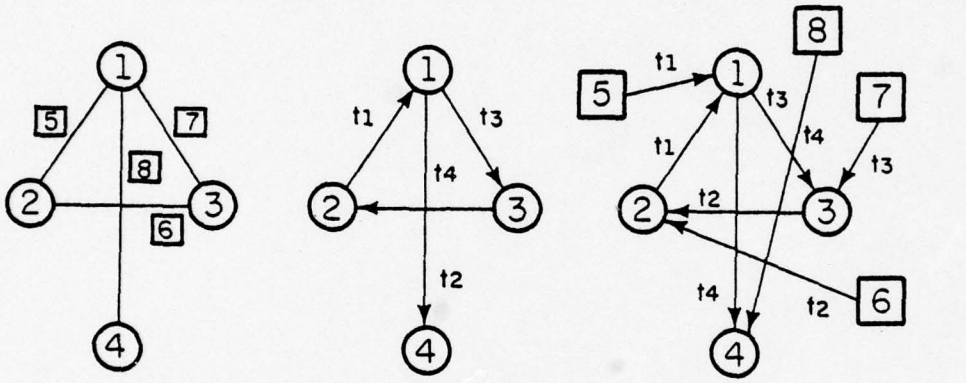


Figure 4.5 Incorporation of Link Failures into System Diagnostic Graph

Figure 4.5(a) shows the communications graph for the system with the links labelled for identification. In Figure 4.5(c) the additional nodes 5, 6, 7 and 8 correspond to faults in the links 1 - 2, 2 - 3, 3 - 1, and 4 - 1 respectively. Since test t_1 in Figure 4.5(b) is essentially a test administered by node 2 on node 1, it requires that link 1 - 2 be up. Hence failure of link 1 - 2 can invalidate test t_1, a fact reflected in the modified diagnostic graph by an arc directed from new node 5 to node 1, labelled test t_1.

In most of the discussion to follow, however, faults will refer to node faults, with link faults not explicitly considered in the system diagnostic graph. They will be taken into account implicitly by points (i) through (v) described above. This allows one to proceed and obtain system graphs which contain the required number of nodes and for which certain links are required from testing, reconfiguration and reusability conditions. The remaining interconnections between the nodes may be added according to architectural and performance requirements.

4.6. Fault Models

In determining what fault models are to be used in a study of this nature two basic approaches come to mind. On the one hand, one could take a probabilistic approach, assuming that failure probability for any given unit follows a certain pattern with time and that a system fails when the basic system is no longer contained as a subgraph in the existing system. An approach of this type has been taken by Abraham and Metze [23]. The work aimed at determining the reliability and the performance of known system configurations and a comparison of measurable parameters for the roving diagnosis approach with those for classical redundancy approaches. Their

results indicate that a 3 - 3 bipartite connection of six nodes is almost as reliable, though far superior in utilization, as a hybrid redundant scheme involving triplication and three spares. (The basic system required two nodes of identical type in their example.)

On the other hand, a more deterministic approach could be taken, where the basic assumption would be that no more than d faults in the system could occur before the existence of faults in the system is detected. This detection may be a result of periodic testing of the system or of irregular operation of some unit in the system. In practice, it may be better to determine the frequency of testing based on a chosen value of d and the mean time between failures of units in the system.

The deterministic approach is the one that will be taken in the next chapter. While the deterministic approach leaves no margin of uncertainty in system designs for a given fault model, it tends to yield designs which are tolerant with a high probability to more than the designed number of faults.

5. ROVING DIAGNOSIS

5.1. Introduction

The previous chapter introduced the concept of roving diagnosis as a practical and viable means of performing diagnosis in large distributed digital systems. Any diagnosis strategy implies some basic conditions that must be met by a system employing that strategy. Thus, where testing is concerned, there are certain configurations of the diagnostic graph that lend themselves to convenient roving diagnosis. Similarly, reconfiguration and reusability are governed by the organization of the communications graph of the system.

The object of this chapter will be to determine first how a diagnostic graph must look in order that roving diagnosis be possible, and second whether systematic algorithms exist for analyzing systems with known diagnostic graphs. The following chapter will examine more closely the problem of reconfiguration and discuss the practical questions posed by the diagnosis strategy.

5.2. Roving Graphs

Figure 5.1 shows the diagnostic graph for a system which has been modified to include only a subset of the tests, retaining all the invalidating faults for those tests selected. It may be recalled that, merely for convenience, a fault will be considered synonymous with a faulty unit or a faulty node.

Two features may be noted about this graph:

(i) The faults f_1 and f_9 are shown without any tests. (These tests may have existed in the unmodified diagnostic graph, however.)

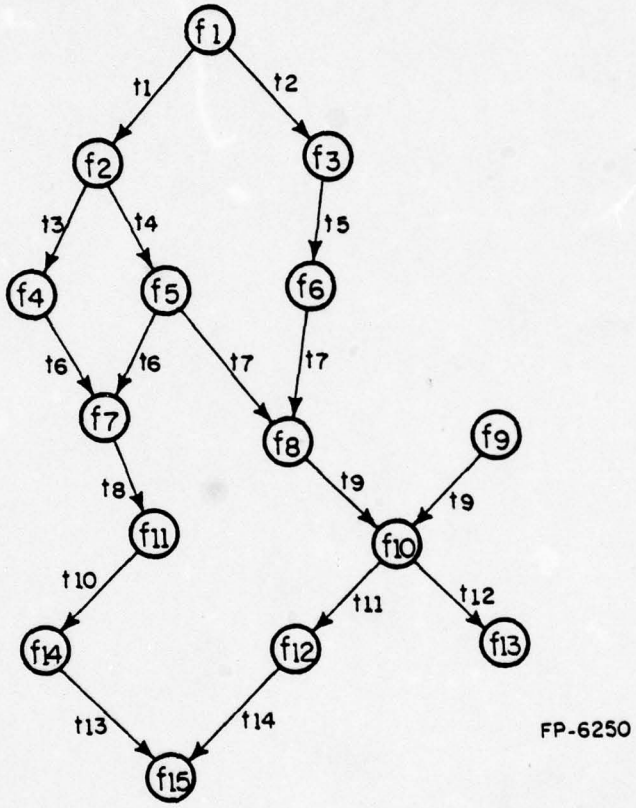


Figure 5.1 A Sample Roving Graph

(ii) If the faults f_1 and f_9 can somehow be shown to be absent from the system, then there exists a systematic sequence of tests which may be used to "rove" through the system. For the graph shown one such sequence could be $\{t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_9,t_{10},t_{11},t_{12},t_{13},t_{14}\}$. By employing this sequence of tests, one can always make sure that a test currently being administered cannot be invalidated, under the assumption that no faults occur during the entire test period. Of course, when some test t_i fails the fault f_i such that $t_i \in t(f_i)$ must have occurred, and the appropriate software routines now can take over to inform the concerned nodes about the fault f_i.

A distinguished node, like f_1 or f_9 in Figure 5.1, which has no incoming arc to it will be referred to as an initial node. A properly ordered test sequence is defined as one in which, for every pair of tests (t_i,t_j) in the sequence such that $t_i \in t(f_i)$ and $t_j \in T(f_i)$, t_i precedes t_j in the sequence. A graph which possesses some initial nodes and in which a properly ordered test sequence can be found which covers all the tests in the graph will be termed a roving graph. A k-roving graph will be defined as a roving graph with k initial nodes. The example of Figure 5.1 is a 2-roving graph.

A roving graph is a form of diagnostic graph because the directed edges represent tests and a relationship exists between the fault detected by a test and faults invalidating that test. Thus, while every diagnostic graph need not be a roving graph, a subgraph of the diagnostic graph can always be found which is a roving graph. In order for the roving graph to

be meaningful, all the faults that invalidate a test in the diagnostic graph must invalidate that test, if it appears, in the roving graph.

A roving subsystem $S_R = (\mathcal{F}, \mathcal{T}_R, F, G_R)$ of a system $S = (\mathcal{F}, \mathcal{T}, F, G)$ is a subsystem of S such that $\mathcal{T}_R \subseteq \mathcal{T}$, the diagnostic graph for S_R is a roving graph, and if $t_i \in \mathcal{T}_R$ then $t_i \in T_R(f_j)$ for every fault f_j such that $t_i \in T(f_j)$. A minimal roving subsystem of S is a roving subsystem of S which has the least number of initial nodes.

Example 5.1

Figure 5.2(b) and (c) show two roving graphs for a system whose diagnostic graph is shown in Figure 5.2(a). The two roving graphs are 3-roving and 1-roving respectively. It may be verified by inspection that the graph in Figure 5.2(c) is the diagnostic graph of a minimal roving subsystem of the original system. □

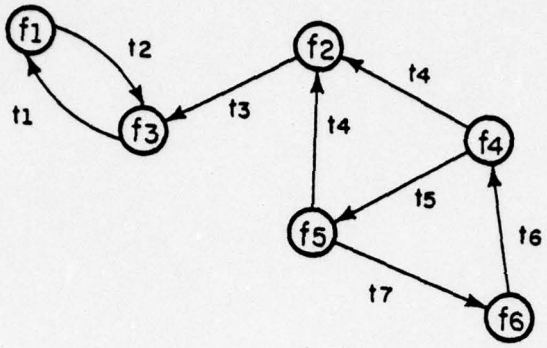
Consider the roving graph as a simple directed graph. Define the relation $\mathcal{R}$ between two nodes f_i and f_j in the graph such that $f_i \mathcal{R} f_j$ if and only if there exists a directed path in the graph from node f_i to node f_j.

Theorem 5.1

A diagnostic graph is a roving graph if and only if the binary relation $\mathcal{R}$ on its set of nodes defines a partial ordering.

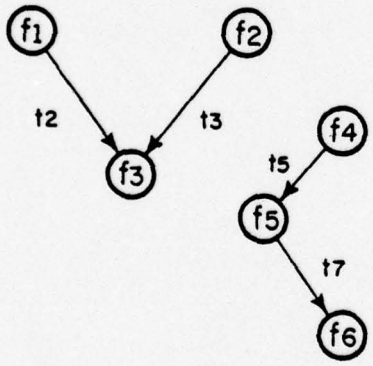
Proof:

Assume that the diagnostic graph is a roving graph. It is easy to observe that for every node f_i in the graph, $f_i \mathcal{R} f_i$, and for nodes f_i, f_j and f_k, if $f_i \mathcal{R} f_j$ and $f_j \mathcal{R} f_k$, then $f_i \mathcal{R} f_k$. Hence $\mathcal{R}$ is reflexive and transitive. Assume the existence of nodes f_i, f_j such that $f_i \mathcal{R} f_j$ and $f_j \mathcal{R} f_i$. By the definition of a roving graph, $f_i \mathcal{R} f_j$ implies that all tests

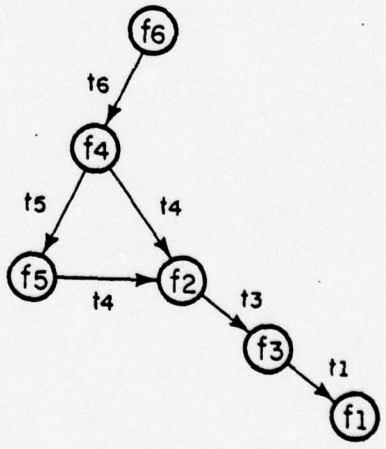


FP-6251

(a) Diagnostic Graph of System S



(b) 3-Roving Graph for S



FP-6252

(c) 1-Roving Graph for S

Figure 5.2 Roving Graphs for a Sample System

belonging to $t(f_i)$ must precede some test t_j belonging to $t(f_j)$. $f_j \mathcal{R} f_i$ implies that all tests belonging to $t(f_j)$ must precede some test t_i belonging to $t(f_i)$. But both t_i and t_j cannot precede each other. Hence if $f_i \mathcal{R} f_j$ then $f_j \mathcal{R} f_i$ cannot be true. Thus $\mathcal{R}$ is antisymmetric besides being reflexive and transitive and hence a partial ordering.

Conversely if $\mathcal{R}$ is a partial ordering, then there must be a "smallest" node f_s such that there is no node f_t satisfying $f_t \mathcal{R} f_s$. It is hence possible to obtain a sequence of nodes Ψ where the i^{th} element in the sequence is the "smallest" node after eliminating the first i - 1 nodes from the diagnostic graph. From the sequence Ψ one can get a sequence of sets of tests where the i^{th} set is the set of complete tests for the i^{th} element in Ψ. By the construction procedure, if $t_i \in t(f_i)$ and $t_j \in T(f_i)$, then f_i must be "smaller" than f_j and hence t_i precedes t_j *in the sequence*. Thus the sequence of tests obtained as described must be a properly ordered test sequence, implying that the diagnostic graph is a roving graph. Q.E.D.

The proof of the above theorem affords a means of determining an appropriate sequence of tests given a roving graph. It must be noted that nothing has been mentioned so far about the tests for the initial nodes themselves. It has been implicitly assumed that they are fault-free. Discussion on this aspect will be delayed until the end of the chapter.

Example 5.2

Consider the roving graph of Figure 5.3. The "smallest" node initially is f_1. After removing f_1 from the system there are three candidates for the next "smallest" node, namely, f_2, f_3 and f_4. Any one of them is chosen arbitrarily. Continuing in this manner, a sequence of nodes Ψ can be obtained as $(f_1, f_2, f_5, f_3, f_4, f_6, f_7, f_8, f_9, f_{10})$. A properly ordered sequence

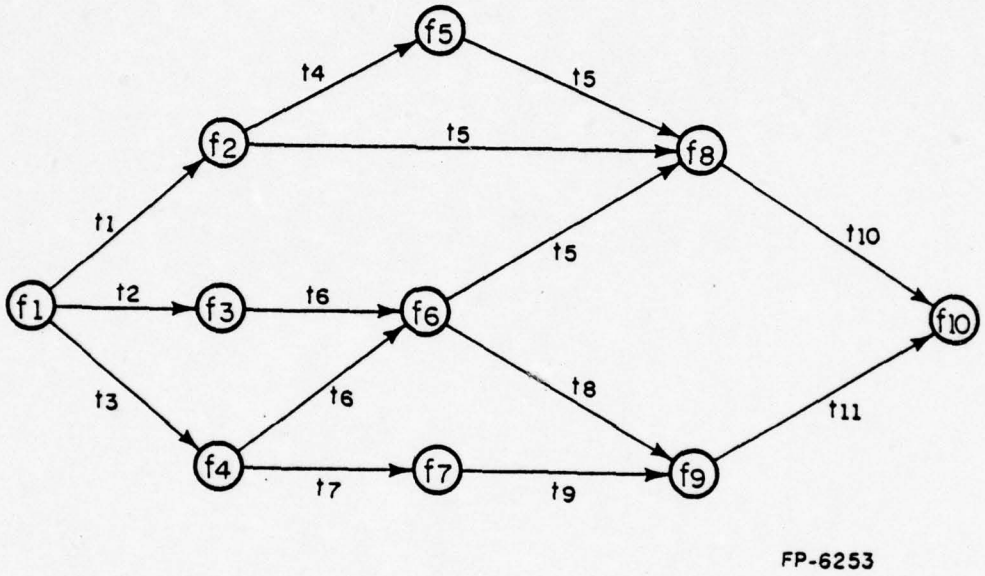


Figure 5.3 Roving Graph for Example 5.2

of tests can thereafter be obtained as $(t_1, t_4, t_2, t_3, t_6, t_5, t_8, t_9, t_{10}, t_{11})$. □

5.3. Minimization of Test Time

The time required to complete the tests may often be important. By minimizing the system test time, the time available for computation is increased, while minimizing the possibility of failures occurring during testing.

Every test for a fault in the system roving graph may be associated with a certain finite time which represents the time required for that test to be carried out. Various factors enter the picture when the total system testing time is to be computed. It may be possible to carry out two tests simultaneously in spite of the fact that both these tests are invalidated by the same fault. Further, all the tests for a given fault need not be performed in order to ensure the absence of faults in the system; any one complete test will suffice. Also, if two faults invalidate some test, it may be either because both the testing units cooperate during the entire test or because one of the testing units carries out one part of the test and the other the remaining part. In view of these considerations, it is not possible to give general algorithms which minimize the total test time for the system. A knowledge of the system characteristics may aid in generating a minimal test-time algorithm for the system, however.

As an example consider a STPF system as shown in Figure 5.4. It is assumed that all units which test a given unit cooperate in performing the test, but that any tests which are invalidated by faults already tested, may be started simultaneously.

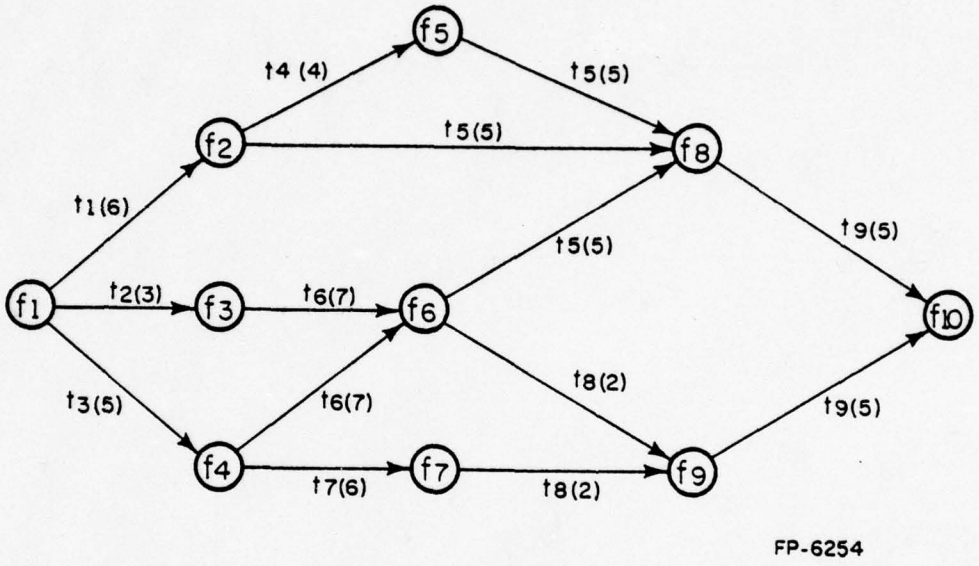


Figure 5.4 An STPF Roving Graph

The figures in parentheses accompanying the test labels indicate the time required for performing the test. One could now take an approach similar to that used in PERT [24] to determine the time for completion of testing.

The length of any path (i.e., the sum of the time durations of the tests on that path) from f_1 to f_i represents a lower bound on the elapsed time, measured from the start of testing before fault f_i has been tested for and before tests invalidated by f_i may be started. It is convenient to associate numbers (times) with vertices as follows:

$$L(f_1) = 0$$

$$L(f_i) = \max \{\ell(P)\} \text{ for } i \neq 1$$

where $\ell(P)$ denotes the time-length of path P and where the maximum is taken over all paths from f_1 to f_i.

Testing is completed when node f_{10} has been tested for. Let $L(f_{10})$ be the time of completion. Define

$$M(f_{10}) = L(f_{10})$$

$$M(f_i) = L(f_{10}) - \max \{\ell(p)\} \text{ for } i \neq 10$$

where $\ell(p)$ is the length of a path from f_i to f_{10} and the maximum is taken over all such paths. The effect of these operations is illustrated in Figure 5.5, where for every node, the pair $(L(f_i),M(f_i))$ is shown.

The <u>critical path</u> for the graph is (f_1,f_4,f_6,f_8,f_{10}). The non-critical nodes represent those faults for which there is a latitude for scheduling their tests, the amount of latitude or <u>slack</u> being given by the difference $M(f_i) - L(f_i)$ for the node.

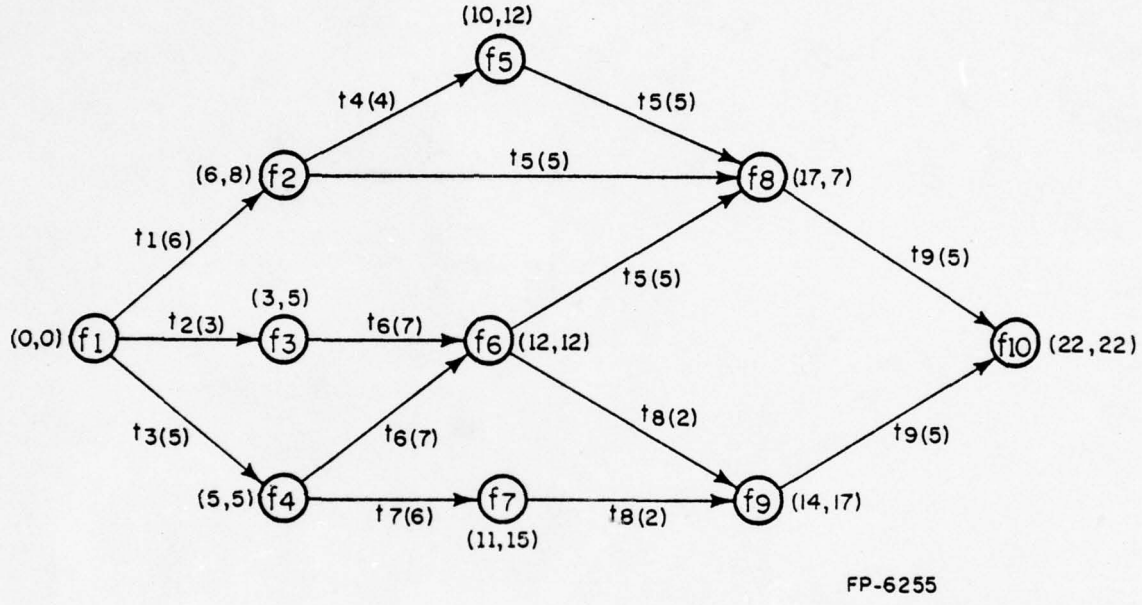


Figure 5.5 Rowing Graph of Figure 5.4 with Labelled Nodes

The assumption made above regarding the possibility of simultaneous execution of those tests which have a common invalidating fault, may often be unreasonable. Figure 5.6 shows a feasible schedule, obtained by trial and error, which does not make this assumption.

The test sequence used to obtain this 30-time unit schedule is $\{t_2,t_3,t_1,t_4,t_5,t_6,t_7,t_8,t_9\}$, with each test being performed at the earliest possible time.

5.4. Determination of Roving Graph

Once a roving graph has been obtained, as shown in the previous sections, systematic techniques exist to determine an appropriate test sequence for the system in order to perform roving diagnosis. This section attempts to find suitable algorithms for determining one or more roving graphs given the diagnostic graph of the system.

Again it is convenient to consider the simpler SMPT and STPF cases before going on to the general case. Further, the nature of the roving graph will differ depending on the number of faults that may have occurred in the system before testing begins. Define a d-fault k-roving graph to be one in which there are k initial nodes and all the faults in the system may be identified by an appropriate sequence of tests as long as not more than d faults that could have occurred in the system.

It may be observed here that if a system has a diagnostic graph which embeds a d-fault roving graph, then the system must be d-fault testable by roving diagnosis. (The initial nodes must be diagnosable themselves, of course.)

An invalidating set of faults $I(f_i)$ for a fault f_i is defined as a set of faults such that $t(f_i) \subseteq T(I(f_i))$.

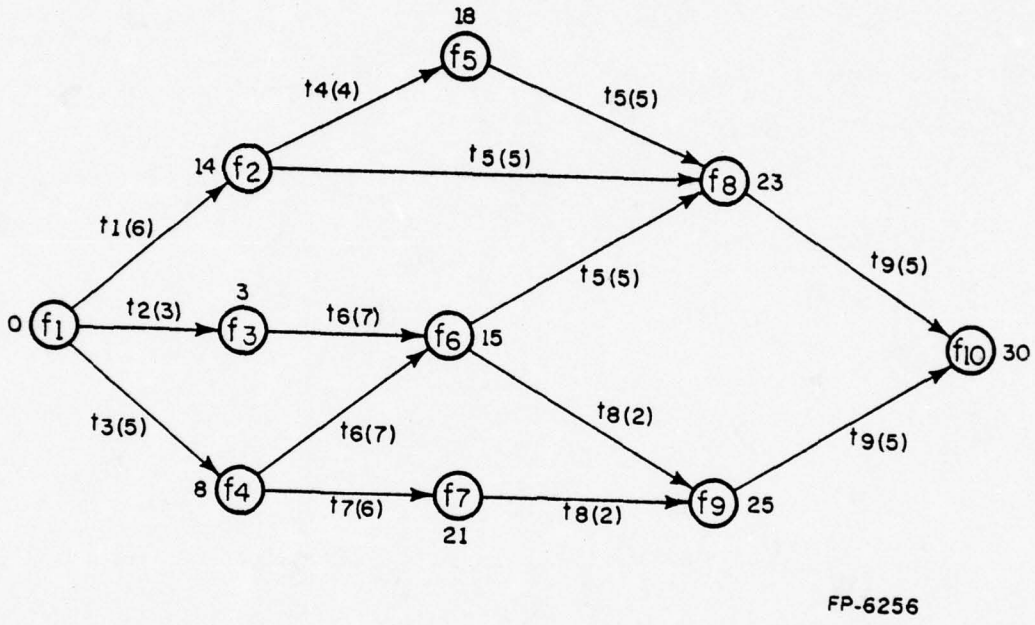


Figure 5.6 Modified Schedule for Roving Graph of Figure 5.4

Theorem 5.2

A roving graph is a d-fault roving graph if and only if the smallest invalidating set for every non-initial node has a cardinality of at least d in the roving subgraph.

Proof:

Let some invalidating set of faults F^g for a fault f_i be such that $|F^g| < d$. Since $t(f_i) \subseteq T(F^g)$ the fault $F^g \cup f_i$ is undiagnosable, even though $\{F^g \cup f_i\} \subseteq F(d)$.

Conversely, assume that every invalidating set of faults in the roving graph has a cardinality of at least d. (This automatically implies the existence of at least d complete tests for every non-initial node in the roving graph.) Construct a sequence of nodes Ψ for the graph as described in the proof of Theorem 5.1. By replacing every non-initial node in the sequence by some test for that node, and ignoring all initial nodes, a properly ordered test sequence for the roving graph is obtained. Now assume that a fault f_i is detected by the failure of some test. Consider a test t_j in the previously obtained test sequence such that $t_j \in T(f_i)$. Let $t_j \in t(f_j)$. Replace t_j in the sequence by a test $t_k \in t(f_j)$ such that $t_k \notin T(f_i)$. Such a test t_k must exist if the cardinality of the invalidating set for f_j is greater than d. If the cardinality equals d then test t_k does not exist only if all the faults in the invalidating set for f_j have occurred, in which case f_j could not have occurred (there being no more than d faults in the system) and need not be tested for. Thus a valid properly ordered test sequence may always be obtained for the system to perform roving diagnosis as long as there are no more than d faults. Q.E.D.

5.5. Roving Graphs for SMPT Systems

Diagnostic graphs of SMPT systems are characterized by a unique test label for every arc in the graph. This fact simplifies the procedure for finding roving graphs for such systems.

For a system having $|\mathfrak{F}| = m$ fault nodes, the adjacency matrix, A, is defined as the $m \times m$ matrix such that an element a_{ij} satisfies

$$a_{ij} = \begin{cases} 1, \text{ if } T(f_i) \cap t(f_j) \neq \emptyset \\ 0, \text{ otherwise.} \end{cases}$$

Thus $a_{ij} = 1$ implies that there is an arc directed from node f_i to node f_j in the diagnostic graph for the system. By multiplying the adjacency matrix by itself m times, an $m \times m$ matrix is obtained in which the j^{th} element of the i^{th} row is a 1 if and only if there exists a directed path in the diagnostic graph from node f_i to node f_j. Call this matrix the connectivity matrix, C.

Theorem 5.3

There exists a 1-roving graph for every SMPT system whose connectivity matrix has at least one row i with $\bigcap_{j \neq i} c_{ij} = 1$.

Proof:

If $\bigcap_{j \neq i} c_{ij} = 1$ then $c_{ij} = 1$ for every element in the row i except the (i,i) element. By the construction of the connectivity matrix, it is clear that there must be a directed path from node i to every other node in the diagnostic graph for the network. Let f_g be an arbitrary node in the network. If the path from f_i to f_g passes through p other nodes in the order $f_i - f_1 - f_2 - \ldots - f_{p-1} - f_p - f_g$ then by the SMPT nature of the graph, any test sequence which includes tests $t(f_1) \cap T(f_i)$, $t(f_2) \cap T(f_1)$, ..., $t(f_p) \cap T(f_{p-1})$, $t(f_g) \cap T(f_p)$ in that order will be properly ordered with

respect to these nodes. By treating f_i as an initial node and trivial extension of the above argument to the other nodes in the graph, the theorem is proved. Q.E.D.

The following algorithm indicates a systematic method for obtaining a roving graph for any SMPT system.

Algorithm 5.1

Step 1: Determine the connectivity matrix, C, for the SMPT system.

Step 2: For each row i in the matrix, determine the sum $\sum_j c_{ij}$. Call it s_i.

Step 3: Find the row p, such that $s_p = \max(s_i)$. Ties are broken arbitrarily. Add f_p to the set of initial nodes.

Step 4: Modify the connectivity matrix, C, eliminating the k^{th} row and column, where $k = p$ or $c_{pk} = 1$. (Eliminate all rows and columns for which k satisfies the above property.)

Step 5: If the resulting matrix is non-null, go back to Step 2. If not, mark all the initial nodes found in the previous iterations. Call the trivial unconnected graph formed by these nodes the graph R.

Step 6: For any marked node in R, say f_i, determine all nodes f_j such that $T(f_i) \cap t(f_j) \neq \emptyset$ and f_j does not already belong to R. Add to R all nodes found thus along with the arcs from node f_i to these nodes with the appropriate test labels.

Step 7: Mark all nodes found in Step 6 and unmark f_i. If there are no marked nodes left in R, then go to Step 8, or else to Step 6.

Step 8: The graph R is a roving graph for the given SMPT system. □

Theorem 5.4:

The graph R obtained by Algorithm 5.1 is a roving graph.

Proof:

The proof is again by construction. At the end of Step 5, define a sequence of tests, ψ, which is initially null. Modify Step 6 as follows:

Step 6: For any marked node in R, say f_i, determine all nodes f_j such that $T(f_i) \cap t(f_j) \neq \emptyset$ and f_j does not already belong to R. Append the tests in $T(f_i) \cap t(f_j)$ to the end of ψ for all the relevant faults f_j. Add to R all nodes f_j found thus along with the arcs from f_i to these nodes labelled appropriately.

Each time a test is appended to ψ in Step 6, it is clear that it can be invalidated only by an initial node or by some other node whose tests have already preceded it in the sequence ψ. Hence the sequence ψ is a properly ordered test sequence, which covers all the tests in R. By the definition of a roving graph R must be a roving graph. Q.E.D.

Theorem 5.5

The roving graph R obtained by Algorithm 5.1 has the smallest number of initial nodes for all possible roving graphs of the system.

Proof:

Assume that two of the initial nodes found in Step 3 of the algorithm are f_g and f_h, with f_g being found before f_h. If f_h were accessible from f_g, i.e. if a path exists from f_g to f_h, then the row and column corresponding to f_h would be eliminated in Step 4 when f_g is an initial node. If, on the other hand, f_h were accessible from f_g, then every node accessible from f_g would also be accessible from f_h implying that $\sum_j c_{hj} > \sum_j c_{gj}$. Hence f_h would have been chosen in step 2 before f_g.

Thus, it is proved that f_g and f_h are inaccessible from each other. Similarly, by taking all the initial nodes found in the algorithm pairwise, and applying the above arguments it is easy to see that the set of initial nodes in R is a set in which there does not exist a directed path between any two nodes. Hence this set of nodes must serve as initial nodes for every roving graph of the system. Q.E.D.

While the roving graph obtained in Algorithm 5.1 has the least number of initial nodes, there may be other graphs which are d-fault-testable for a larger d. It may be observed that the Algorithm 5.1 leads to a forest of trees having the initial nodes as the set of roots. Hence the roving graph thus obtained can be no better than a 1-fault k-roving graph, where k = number of initial nodes. The example which follows illustrates a heuristically derived 2-fault 3-roving graph.

Example 5.3

A roving graph of an SMPT system whose adjacency matrix is shown in Table 5.1 is shown in Figure 5.7(a).

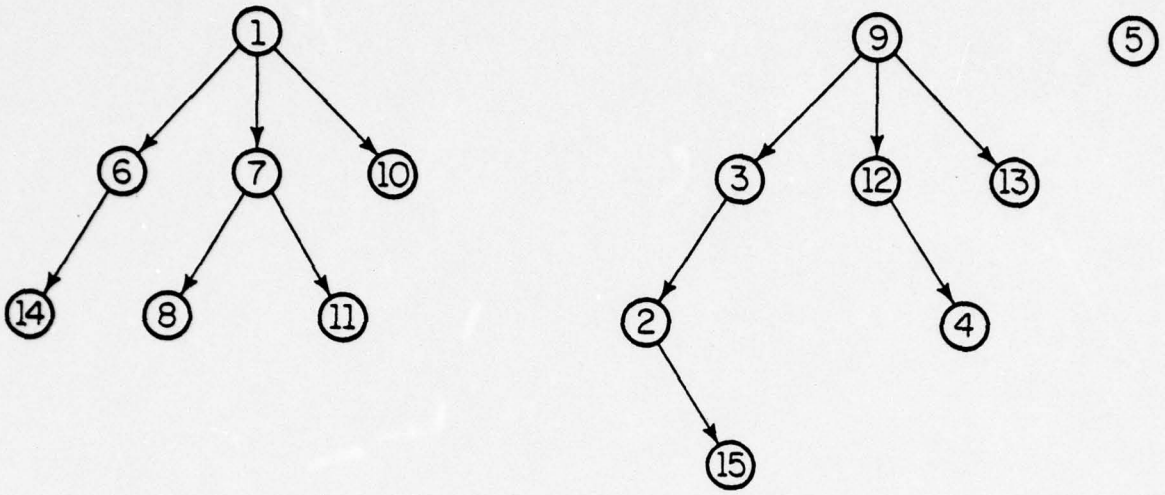
Figure 5.7(b) illustrates how the roving graph obtained by using Algorithm 5.1 on an SMPT system may be modified in order to obtain another roving graph which is now testable for a greater number of faults. The dotted lines indicate those test arcs which exist in the original diagnostic graph but are not necessary in the roving graph. It is also clear that since there are some non-initial nodes which have no more than two tests, there cannot exist a 3-fault 3-roving graph for the system. □

5.6 Roving Graphs for STPF Systems

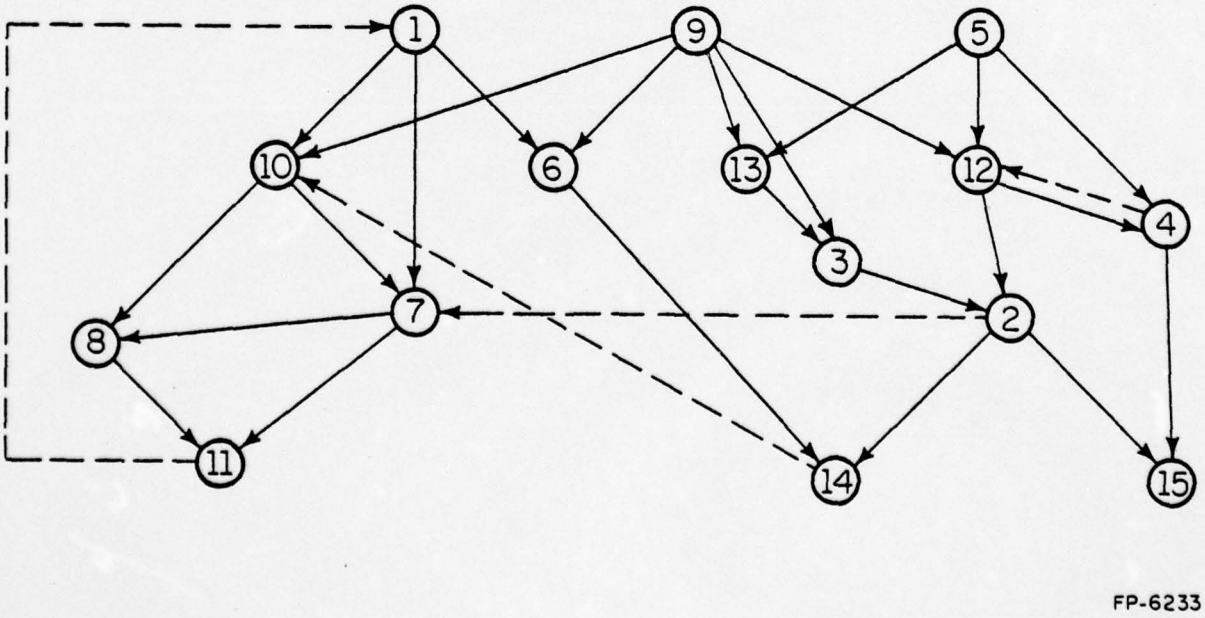
The fact that STPF systems have one and exactly one test per fault leads to two immediate important observations:

Table 5.1 Adjacency Matrix for the System

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
2	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
5	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
9	0	0	1	0	0	1	0	0	0	1	0	1	1	0	0
10	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



(a) A Roving Graph for the System Using Algorithm 5.1



(b) 2-Fault 3-Roving Graph for the System

Figure 5.7 Illustration of Roving Graphs for an SMPT System

(i) There must be at least one cycle in the diagnostic graph of the system. (The implicit assumption is that the system is self-testing as defined in Chapter 2. This assumption is reasonable in view of the fact that roving diagnosis aims at eliminating fault-free hard-core for a system.)

(ii) The roving graph for the system can be no better than a 1-fault roving graph, unless it is a degenerate n-roving graph where $|\mathfrak{F}| = n$.

Algorithm 5.2 indicates how a 1-fault 1-roving graph can be determined, if one exists, in an STPF system.

Algorithm 5.2

Step 1: Determine all the directed cycles in the diagnostic graph for the system. (This may be done by using the variable adjacency matrix of Danielson [25].)

Step 2: Let p be the number of cycles in the system graph, and let y_i, $i = 1,2,\ldots,p$ be the set of nodes involved in the i^{th} cycle. Determine the intersection $\bar{y} = \bigcap_i y_i$.

Step 3: If $\bar{y}$ is empty, then there does not exist a 1-fault 1-roving graph for the system. Otherwise, the roving graph R for the system is obtained by choosing an arbitrary node $f_0 \in \bar{y}$ and deleting from the diagnostic graph for the system, all arcs labelled with the tests belonging to $t(f_0)$. □

Lemma 5.6

If $\bar{y}$ is empty in Step 3 of Algorithm 5.2 then a 1-fault 1-roving graph does not exist for the system.

Proof:

Consider two cycles in the system diagnostic graph whose node sets are y_1 and y_2 and $y_1 = y_2$. Consider the node $f_i \in y_1 \cup y_2$ such that $f_i \notin y_1 \cap y_2$. Without loss of generality, it may be assumed that $f_i \in y_1$. By removing node f_i and its associated test links from the system the cycle with node set y_1 is eliminated from the system but that with node set y_2 still remains, because none of arcs and nodes involved in the latter are affected by the removal of f_i. Hence a node belonging to $y_1 \cap y_2$ must be removed in order to ensure the elimination of both cycles.

By extending the argument for all cycles in the system diagnostic graph the lemma is proved.

Theorem 5.6

The graph R obtained by Algorithm 5.2 is a roving graph for the STPF system.

Proof:

Consider a node belonging to all the cycles in the system diagnostic graph. Every cycle must have one incoming edge to this node. By deleting all the incoming edges to this node, all the cycles in the system graph must be eliminated. The removal of all such incoming edges essentially represents the removal of all complete tests for the chosen node (in this case, one). This implies that in the modified graph, this node will be an initial node. Since every node in the rest of the system has a complete test, and since no cycles remain, there can be no more initial nodes in the system and a properly ordered test sequence exists for all faults in the system. Thus the resulting graph is a 1-fault 1-roving graph. Q.E.D.

The proofs of the above lemma and theorem indicate how one would approach the problem of finding a 1-fault k-roving graph problem for an STPF system with a minimal k.

Algorithm 5.3

Step 1: Determine all the directed cycles in the diagnostic graph for the system.

Step 2: List only the nodes involved in the cycles and eliminate all cycles which involve the same nodes as some other cycle.

Step 3: (Optional) Eliminate all cycles a subset of whose nodes is involved in some other cycle.

Step 4: Construct a $p \times m$ matrix M where the p rows correspond to the p cycles after performing steps 2 and 3 and $m = |\mathfrak{F}|$ the number of fault nodes in the system. The elements of the matrix are defined by

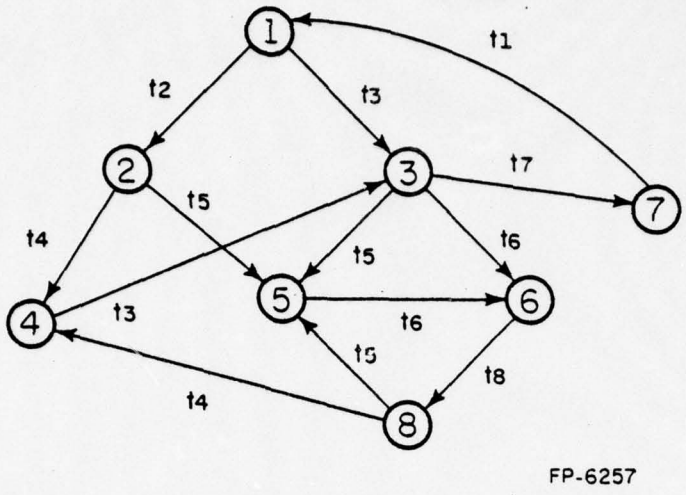
$$m_{ij} = \begin{cases} 1, \text{ if node } j \text{ is contained in cycle } i \\ 0, \text{ otherwise} \end{cases}$$

Step 5: Determine the smallest set of nodes, I, which covers all the rows of the matrix M. (This is simply the classical covering problem and may be solved using the Petrick function [26] approach.)

Step 6: The 1-fault roving graph for the system is obtained by making I the set of initial nodes and eliminating from the diagnostic graph for the system, all arcs corresponding to complete tests for these nodes. □

Example 5.4

The system whose diagnostic graph is shown in Figure 5.8(a) is an STPF system having the following cycles: (i) 1-3-7-1 (ii) 3-6-8-4-3



(a) Diagnostic Graph for an STPF System

Cycle \ Node	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
1-3-7-1	1	0	1	0	0	0	1	0
3-6-8-4-3	0	0	1	1	0	1	0	1
5-6-8-5	0	0	0	0	1	1	0	1

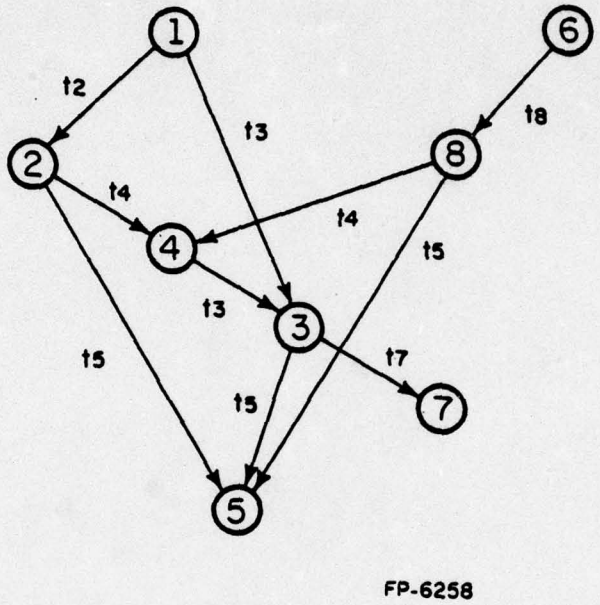
(b) M-Matrix for the System

Figure 5.8 Illustration of a Roving Graph for an STPF System

AD-A069 770 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 9/2
DIAGNOSIS, SELF-DIAGNOSIS AND ROVING DIAGNOSIS IN DISTRIBUTED D--ETC(U)
SEP 78 R K NAIR DAAB07-72-C-0259
UNCLASSIFIED R-823 NL

2 OF 2
AD
A069770

END
DATE
FILMED
7-79
DDC



(c) A 1-Fault 2-Roving Graph for the System

Figure 5.8 Illustration of a Roving Graph for an STPF System (Continued)

(iii) 3-5-6-8-4-3 (iv) 1-2-4-3-7-1 (v) 5-6-8-5. All other cycles may be decomposed into two or more of the above cycles.

By Step 3 of Algorithm 5.3, the cycle (v) is contained in cycle (iii) and hence the latter may be eliminated. Cycle (iv) may be similarly eliminated. The remaining cycles and their relationship to the fault nodes are depicted in the M-matrix of Figure 5.8(b).

The Petrick function for the system is

$$(1 \vee 3 \vee 7)(3 \vee 6 \vee 8 \vee 4)(5 \vee 6 \vee 8) = 16 \vee 18 \vee 67 \vee 87 \vee 35 \vee 36 \vee 38 \vee 145 \vee 475$$

Each term in the sum-of-products expression above indicates the initial nodes in a possible roving graph for the system.

Figure 5.8(c) shows a sample 1-fault 2-roving graph for the system. Note that there are 6 other possible 2-roving graphs for the system and that there is no 1-roving graph for the system.

A properly ordered sequence of tests for the system would be $\{t_2, t_8, t_4, t_3, t_5, t_7\}$. □

5.7 Roving Graphs for General Systems

While some of the techniques mentioned in the previous two sections could be applied with success in the general case, they would not always lead to the best roving graphs for the system. Various criteria may be used in choosing the best roving graph:

(i) the graph with the least number of initial nodes,

(ii) the graph which is maximally fault testable

(iii) a reasonable combination of both (i) and (ii).

Algorithm 5.4 may be used to determine the roving graph of a system with the least number of initial nodes.

Algorithm 5.4

Step 1: Determine all the directed cycles in the diagnostic graph for the system.

Step 2: List only the test edges involved in the cycles and eliminate all cycles which involve the same tests as some other cycle.

Step 3: (Optional) Eliminate all cycles a subset of whose tests are involved in some other cycle.

Step 4: Construct a $q \times n$ matrix N where the q rows correspond to the q cycles after performing Step 3 and $n = |\mathcal{T}|$ the number of complete tests in the system. The elements of the matrix are defined by

$$n_{ij} = \begin{cases} 1, & \text{if test } j \text{ is included in cycle } i \\ 0, & \text{otherwise} \end{cases}$$

Step 5: Determine all the irredundant set of tests, J_i which cover all the rows of matrix N. (Note that an irredundant set need not be the smallest set. An irredundant set of tests is one in which removal of any element of the set leaves some row uncovered.)

Step 6: For each set of tests, J_i, determine the largest set of fault nodes I_i such that $t(I_i) \subseteq J_i$.

Step 7: If I_m is the set or nodes such that $|I_m| = \min_i |I_i|$ then the roving graph for the system is obtained by deleting all edges corresponding to tests in J_m (corresponding to I_m) from the diagnostic graph, and making I_m the set of initial nodes. □

Theorem 5.7

The procedure described in Algorithm 5.4 leads to a roving graph which has the least number of initial nodes.

Proof:

Assume that there exists a roving graph which has fewer initial nodes, I_y. The set of complete tests for these nodes must be one which covers all the q cycles of the diagnostic graph. Further there must exist a subset of this set of tests which is an irredundant set covering all the cycles. Let this set be J_x. But since Algorithm 5.4 finds all the irredundant sets in Step 5, J_x must also be found. Consider the largest set of fault nodes I_x such that $t(I_x) \subseteq J_x$. Since $J_x \subseteq t(I_y)$, clearly $t(I_x) \subseteq t(I_y)$ and hence $I_x \subseteq I_y$. By making I_x a set of initial nodes and by eliminating all tests in J_x, all cycles in the diagnostic graph are broken and hence a roving graph is obtained. But since $|I_x| \leq |I_y|$, it contradicts the original assumption that $|I_y| < \min_i |I_i|$ and the theorem is proved. Q.E.D.

Given that the roving graph for a system be d-fault testable, the following algorithm determines a roving graph which has the least number of initial nodes, assuming that no two tests for a fault are invalidated by the same fault.

Algorithm 5.5

Step 1: Modify the diagnostic graph for the system by eliminating those edges corresponding to t_i where $t_i \in t(f_i)$ and $|t(f_i)| < d$.

Step 2: Perform Steps 1 through 5 of Algorithm 5.4, treating the modified graph as the diagnostic graph for the system.

Step 3: For each set of tests, J_i, found above, determine the largest set or fault nodes $F^i = \{f_1, f_2, \ldots, f_{p_i}\}$ such that $|t(f_j) - J_i| < d$ for all $j = 1,2,\ldots,p_i$. (This ensures that all the other faults in the graph have d or more complete tests.) Set J_i to $J_i \cup t(F^i)$.

Step 4: If F^m is the set of nodes such that $|F^m| = \min_i |F^i|$ then the roving graph for the system is obtained by deleting all edges corresponding tests in J_m from the diagnostic graph. All nodes which do not have an incoming arc as a result of this operation are made initial nodes. □

The only restrictive point about the above algorithm is that it requires that there should be no two faults f_i, f_j such that $|T(f_i) \cap t(f_j)| > 1$. Actually, in practice, one may very well be able to carry out the algorithm as described even though the condition is not satisfied and then ensure that the resulting roving graph does not possess any non-initial nodes which have invalidating sets of faults with cardinality smaller than t.

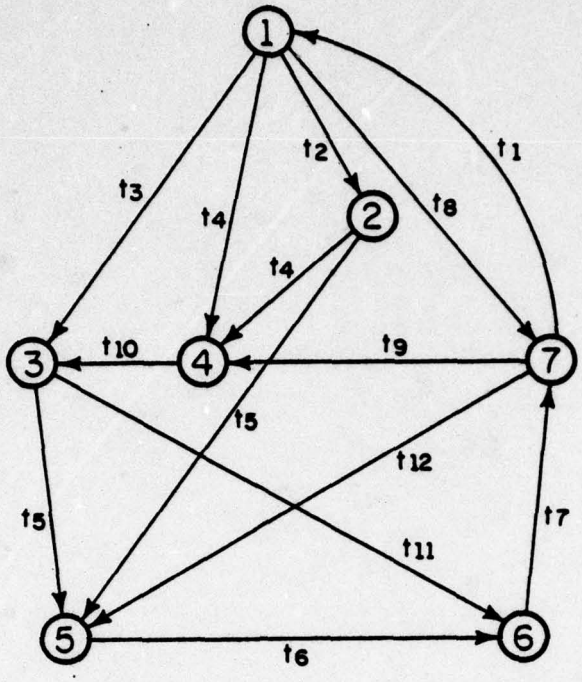
This section is concluded with an example to illustrate the points mentioned.

Example 5.5

Consider a system S whose diagnostic graph is shown in Figure 5.9(a). The set of cycles obtained after performing Step 3 and the *N*-matrix of the system are shown in Figure 5.9(b). The sets of minimal tests are: (1,6,9), (1,6,10), (1,6,11), (1,7), (1,9,12), (1,10,12), (1,11,12), (7,8), (5,8,11,12), (6,8,11).

The corresponding sets of fault nodes (I_i) are given by: (1), (1), (1,6), (1), (1), (1), (1), (7), (5), (6). Thus there are 4 candidates for a single initial node, namely, nodes 1, 5, 6 or 7. Figure 5.9(c) shows a roving graph obtained by eliminating tests t_7 and t_8.

Attempting to obtain a 2-fault roving graph for the system, the modified diagnostic graph is drawn as per Step 1 of Algorithm 5.5 in Figure 5.9(d).

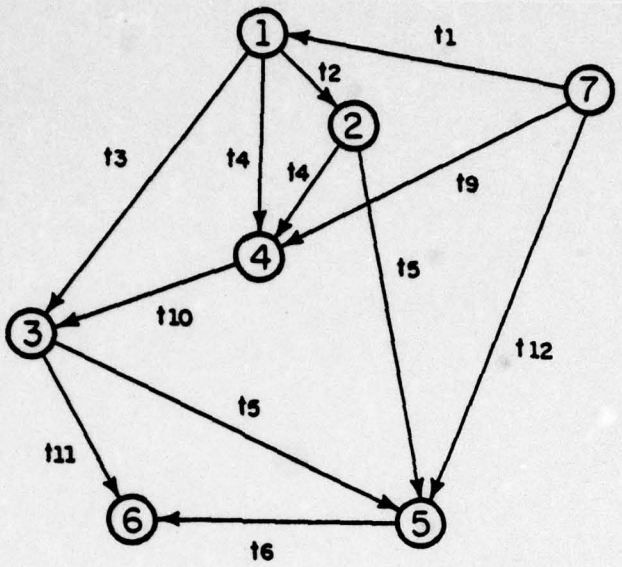


(a) Diagnostic Graph for System S

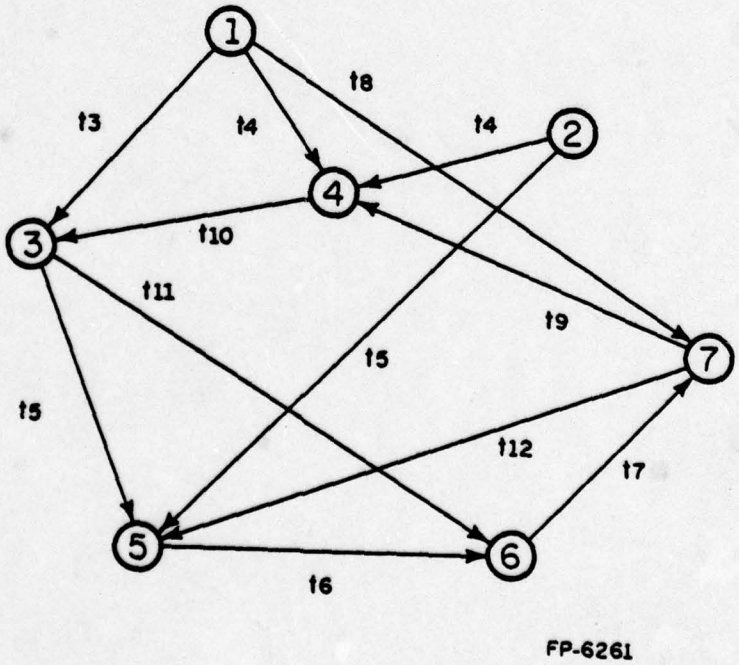
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
Cycle 1	1	0	0	0	0	0	0	1	0	0	0	0
Cycle 2	1	1	0	0	1	1	1	0	0	0	0	0
Cycle 3	0	0	0	0	0	0	1	0	1	1	1	0
Cycle 4	1	0	1	0	0	0	1	0	0	0	1	0
Cycle 5	1	1	0	1	0	0	1	0	0	1	1	0
Cycle 6	1	0	0	1	1	1	1	0	0	1	0	0
Cycle 7	1	0	0	1	0	0	1	0	0	1	1	0
Cycle 8	1	0	1	0	1	1	1	0	0	0	0	0
Cycle 9	0	0	0	0	1	1	1	0	1	1	0	0
Cycle 10	0	0	0	0	0	1	1	0	0	0	0	1

(b) *N*-Matrix for the System

Figure 5.9 "Best" Roving Graphs for the General System of Example 5.5



(c) 1-Fault 1-Roving Graph for System S

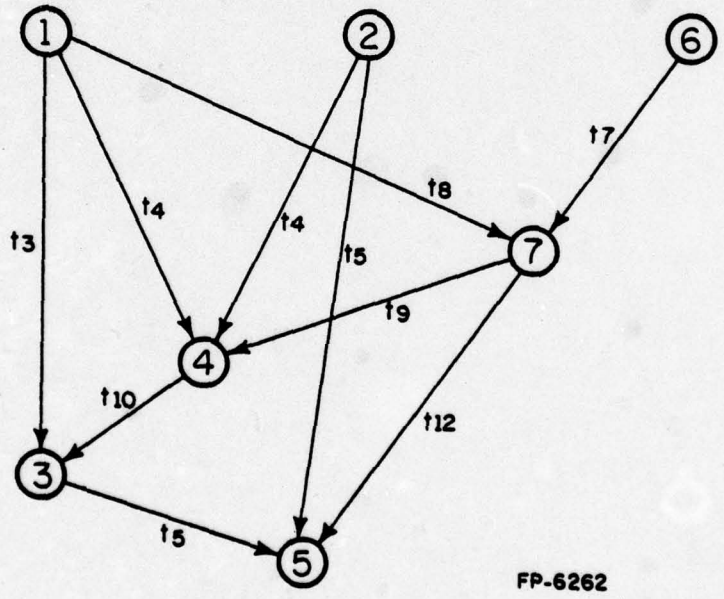


(d) Modified Diagnostic Graph for System S

Figure 5.9 "Best" Roving Graphs for the General System of Example 5.5 (Continued)

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
Cycle 1	0	0	0	0	0	0	1	0	1	1	1	0
Cycle 2	0	0	0	0	1	1	1	0	1	1	0	0
Cycle 3	0	0	0	0	0	1	1	0	0	0	0	1

(e) *N*-Matrix for Modified Graph



(f) 2-Fault 3-Roving Graph for the System

Figure 5.9 "Best" Roving Graphs for the General System of Example 5.5 (Continued)

The new sets of minimal tests are (5,11,12), (6,9), (6,10), (6,11), (9,12), (10,12), (7). The sets of nodes (F^i) of Step 3 in Algorithm 5.5 are: (5,6), (6,4), (6,3), (6), (4,5), (3,5), (7). The 2-fault roving graph having the least number of initial nodes must hence have 3 initial nodes, 1, 2 and 7 or 1, 2 and 6. The latter 2-fault 3-roving graph is shown in Figure 5.9(f).

It may be easily verified that there does not exist a non-degenerate 3-fault roving graph for the system. □

5.8 Diagnosis of Initial Nodes

The initial node used in the earlier sections was actually an aid in determining a properly ordered test sequence for the system. Obviously, the initial nodes are themselves prone to faults and hence may invalidate some tests in the roving graph for the system.

The absence of any tests for the initial faults in the roving graph does not mean that they cannot be tested for in the system. It may be recalled that the roving graph is a subgraph of the diagnostic graph for the system. Hence one may attempt to "come round a complete circle" and diagnose the initial nodes using the nodes which were diagnosed under the assumption that the initial nodes were fault-free.

The first requirement implied by this strategy is that the initial nodes themselves must possess invalidating sets with cardinality at least d (vide Theorem 5.2 in Section 5.4). This would suffice if an initial node under a fault never causes a test which should fail to pass. (This is the case in the Barsi, Grandoni and Maestrini's model [9] but does not hold true for the general model being considered here.)

A second solution may be to use "hard-core" units to check the initial nodes only. This would imply that in order to reduce the complexity of hard-core requirements, firstly, the number of initial nodes would have to be reduced and secondly, the complexity of the initial nodes themselves would have to be reduced in order to make the tests simpler. In any case, the requirement that some sort of global fault-free tester be available in this scheme is somewhat disconcerting.

A better approach would utilize self-checking units for the initial nodes. The hard-core requirement in this case could be simplified to a simple monitor which looks at the output of the checker and sets off an alarm when something goes awry with the initial node. However, even in this scheme, the initial nodes would have to undergo tests periodically, but not quite as frequently as before, because it is impossible to guarantee the exercise of all the necessary inputs during regular operation. Discussions on the design of self checking circuits may be obtained from [27], [28] and [29].

By keeping the initial nodes fairly simple, the self-checking designs of these circuits may be made without undue difficulty. Such an approach is viable and very useful, especially if there is a degree of flexibility allowed in the design of the initial nodes of a distributed network.

Any further reduction in the global requirements for the system will tend to increase the demands on the structure of the network itself. One such approach which is being termed <u>cooperative initial diagnosis</u> will now be described.

Assume that at the start of the testing procedure, a certain set of nodes in the system simultaneously tests an initial node. At the completion of all the tests, these nodes cooperate among themselves to find out how many of the nodes found the initial node to be faulty and how many fault-free. Depending on the result, and on the fault assumption, it can be determined unambiguously whether the initial node is faulty or not.

As an example, Figure 5.10 shows a part of the diagnostic graph for a 2-fault system where f_i is the initial node. The fault nodes f_1, f_2, f_3 and f_4 correspond to the nodes which attempt to diagnose f_i through tests t_1, t_2, t_3 and t_4 respectively. If two or more of these nodes find fault f_i to be absent, then that must indeed be the case, or else there would be three or more faults in the system. If the number is fewer than two, then, for a similar reason, f_i must correspond to an existing fault.

In order that the nodes corresponding to f_1, ..., f_4 communicate among themselves, the communications graph for the system must include arcs between these nodes. In fact, the possibility of faults among these units could imply a complete graph among these nodes for the communications graph.

For a d-fault system, the above mechanism may be directly extended with 2d nodes testing an initial node and a complete graph among these nodes in the communications graph for the system.

There is just one piece of hard-core implied by the cooperative initial diagnosis approach. There must be a mechanism by which all the testing units are instructed to begin testing the initial node. A clock common to these units is one such mechanism.

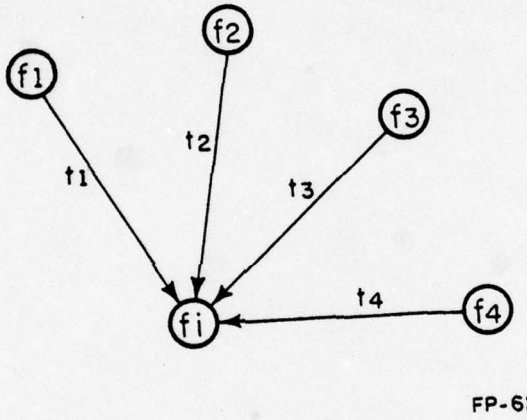


Figure 5.10 Diagnosing the Initial Node f_i

In practice, the intial node diagnosis problem is not as complex as it has been made out to be in this section. Once a testing routine has been completed around the roving graph, if there are no faults found then there is a high probability that the nodes testing the initial nodes are themselves fault-free when they are ready to test the initial node. Hence one may complete the loop by proceeding to test the initial nodes just like any other node. Until a fault is actually detected, this continuous roving around the system is guaranteed to give reliable results. (This procedure may imply a certain minimal length for the loops involved. For example if a cycle of length d is involved in an SMPT system, there is a possibility, though rather remote, that all d units are actually faulty, but are declaring one another to be fault-free.)

The next chapter will consider the problem of reconfiguration and reusability in the system and discuss some practical aspects of roving diagnosis.

6. FURTHER IMPLICATIONS OF ROVING DIAGNOSIS

6.1 Introduction

The previous chapter described the necessary conditions to be satisfied by a system in order that it may be able to detect faults with minimal external help. The system however continues to be useful only if it is aware of the existence of the faults and can resume functioning from a point where the results are known to be unadulterated. The latter problem, called the recovery problem, usually requires the system software to support rollback and hence effect recovery. The former, which will be considered here, implies certain interconnections between the nodes in a system in order that the system may continue operation, although with a possible degradation in performance or throughput.

6.2 Reconfigurability

In Chapter 4, a system was defined as being reconfigurable if all the units in the vicinity of a faulty unit can be reliably informed about the fault. At least one of the units which has test edges directed to the faulty unit must know about the fault as soon as it is detected. (All of them need not know because some units may simply be feeding in the test stimuli, the response to which may be observed by other units.) If all the other units adjacent to the faulty unit are reachable from the unit detecting the fault then the faulty unit may be effectively isolated from the rest of the system. (Reachability here refers to the communications graph. The choice of synonymity between a fault and a faulty node simplifies matters here. Otherwise one would have to refer to the "node containing the detected fault" rather than the "faulty node.")

As mentioned before, the communications graph for the system may be an undirected graph, as in networks with full-duplex links. However, by treating it as a directed graph, results which are more general may be obtained.

The connectivity of a graph is the minimum number of nodes whose removal disconnects the graph. When a graph is disconnected, there exists at least one node which does not have a directed path to some other node in the graph. Hence it follows that a system is reconfigurable to at least χ faults if χ is the connectivity of the system communications graph. There could be a set of faults F^i, $|F^i| > \chi$ such that the system is reconfigurable to F^i, since F^i may not form a cutset of the communications graph.

It is also possible for the system to be reconfigurable to a set of faults which leaves the system disconnected, as shown in the example below.

Example 6.1

Consider the diagnostic graph shown in Figure 6.1. Assume that the communications graph for the system is an undirected graph with edges between the same pairs of nodes having edges in the diagnostic graph. The communications graph is not 1-connected because a faulty node 7 leaves the graph disconnected.

Since node 7 has two complete tests t_4 and t_6 it is possible to detect a fault in the node through node 4 or node 5. Thus both nodes can independently learn of the fault and no more nodes need to be informed about the fault in order to effect reconfiguration.

Hence it is seen that reconfigurability does not necessarily imply connectivity, though connectivity is a sufficient condition for

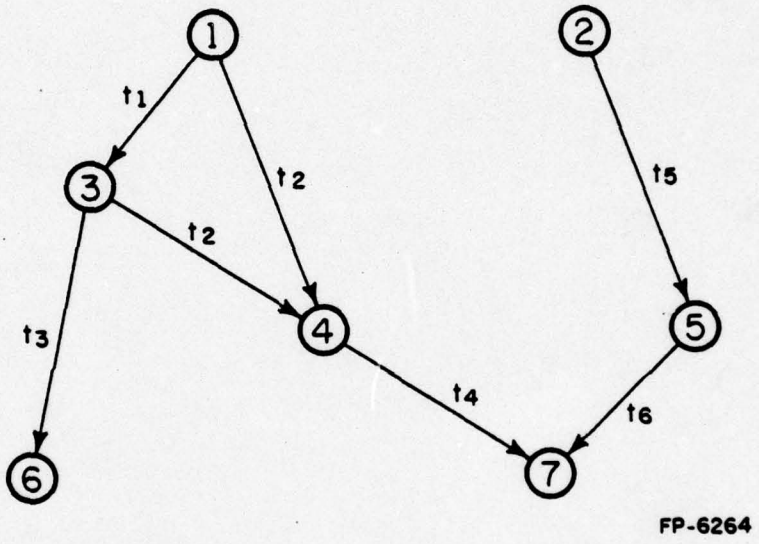


Figure 6.1 A Sample Diagnostic Graph for Example 6.1

reconfigurability. In most cases, a test for connectivity would suffice in order to determine whether the system is reconfigurable or not. But in some cases when the connectivity is not high enough, it may be necessary to look at the roving graph for the system and the nature of implementation of the testing routines in order to ensure reconfigurability. Many graph-theoretic results exist which relate the connectivity of graphs with other characteristics of the graph like the degrees of nodes. Some of these may be obtained from Wilkov [30] or Frank and Frisch [31].

The next important aspect which needs consideration is the time required to effect reconfiguration. It is imperative that the neighbors of the fault know about the fault promptly in order to avoid erroneous results during operation.

The time required to effect reconfiguration is roughly a function of the maximum number of hops (link traversals) that have to be made in order to reach all the units neighboring the fault from the units which know about the fault. In the worst case, this time is proportional to n-2, where n is the number of nodes in the system graph. This will be the case when the fault message has to travel over a Hamiltonian path of the system, as in the case of a single-loop system.

If A is a directed graph with n nodes, then A^k, the graph obtained by k-rotation of A is defined as that in which node f_i, $0 \leq i < n$, is renamed as node $f_{i'}$, where $i' = (i+k) \bmod n$. A graph A which possesses a labelling of its nodes, $f_0, f_1, \ldots, f_{n-1}$, such that $A^k = A$ for $0 \leq k < n$, will be called a symmetric graph. (Two graphs A and B are said to be equal, A = B, if and only if they are identical in all respects.)

Lemma 6.1:

If there is a directed arc from f_1 to $f_{(i+r) \bmod n}$ in a symmetric graph A, then there must be a directed arc from f_j to $f_{(j+r) \bmod n}$ for all j.

Proof:

Assume that some node $f_{(i-k) \bmod n}$ is not the origin of an arc to $f_{(i-k+r) \bmod n}$. The graph A^k obtained by k-rotation of graph A must hence have no arc directed from node f_i to node $f_{(i+r) \bmod n}$. But then $A^k \neq A$ implying that A is not symmetric--a contradiction. Q.E.D.

Symmetric graphs are interesting because in certain cases they are optimal with respect to the time required for reconfiguration. Since the number of hops required to reach some node is a good indication of the time required to send a message to that node, it may be convenient to define H, called the reconfiguration delay, as the maximum number of hops required to reach nodes adjacent to a faulty node from a node detecting the fault. If broadcasting or simultaneous transmission of a fault message is possible from a node then H is a parameter proportional to the time required to effect reconfiguration.

The following is an interesting result applicable to symmetric communications graphs. (Addition will be assumed to be modulo n addition, where n is the number of nodes in the system graph.)

Theorem 6.1

If the communications graph for a 1-fault testable system is symmetric and undirected with connectivity ≥ 2, then the reconfiguration delay (H) ≤ 4.

Proof:

Consider the node f_i which has been found to be faulty by node f_j, where $i = j + k$. Let f_i be connected to some other node $f_{i+p} = f_q$. Since the graph is symmetric there must be an arc between f_j and f_{j+p}. There must also be an arc between f_{j+p} and f_{j+p+k} because if f_j detects f_i

to be faulty then there must be an arc between these nodes in the communications graph. But $j + p \neq k = i + p$ implying that as long as $p \neq k$, two hops, $f_j - f_{j+p} - f_{i+p}$ are sufficient for propagation of fault information.

If $p = k$ then the above operation cannot be performed because $f_{j+p} = f_{j+k} = f_i$ is itself a faulty node. But since the connectivity of the communications graph is at least 2, every node must have degree at least 3, implying that it is possible to find an integer q such that the communications graph possesses arcs $f_j - f_{j+q}$, $f_i - f_{i+q}$ and $f_{i+p} - f_{i+p+q}$. If $j + q = i + p$ then clearly only one hop is required. Otherwise neither f_{i+q} nor f_{i+p+q} can be identical to node f_i implying that they cannot be faulty by the single fault assumption. The following hops hence form a valid fault-free path: $f_j - f_{j+q} - f_{i+q} - f_{i+p+q} - f_{i+p}$.

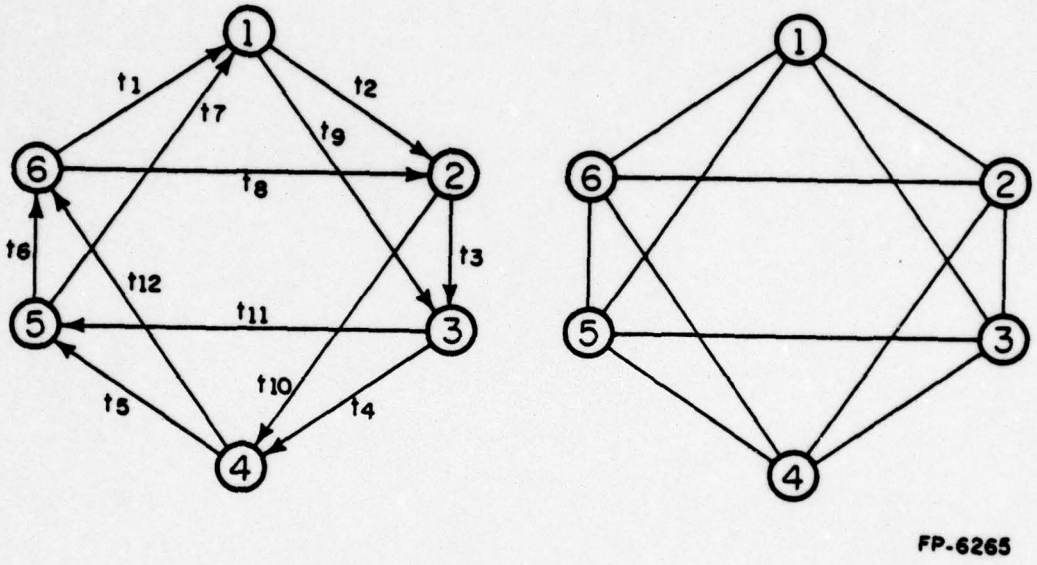
Applying the same arguments to every node connected to f_i, the theorem is proved. Q.E.D.

Example 6.2

Assume that the diagnostic graph and the communications graph for a system are as shown in Figure 6.2. (The initial node is tested by the cooperative testing technique and any node qualifies to be an initial node.)

Let 1 be the initial node with the corresponding roving graph consisting simply of the chain $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$. Assume that node 4 finds node 5 to be faulty. Reconfiguration will be affected when node 4 can inform nodes 6, 1 and 3 about the fault in node 5. The shortest hops in this case are 4-6, 4-6-1 and 4-3 implying a reconfiguration delay of only 2.

For graphs with a small number of nodes, it will be found that the reconfiguration delay ≤ 2. Also for graphs with large number of nodes



(a) System Diagnostic Graph (b) System Communications Graph

Figure 6.2 Sample System for Example 6.2

and high connectivity, the average number of hops to reach the neighbors of a faulty node will be found to be close to 2.

In concluding this section, it may be mentioned that it is conjectured that the interesting properties of symmetric graphs extend also to t-fault testable systems, though the specific bound on the reconfiguration delay may be a function of t.

6.3 Reusability

As defined in Chapter 4 a system is said to be reusable after a fault if the system is reconfigurable after detection of the fault and the communications graph of the reconfigured system has the minimum number of nodes and minimum links between the nodes necessary for useful functioning of the system. The definition implies that there is a basic system which must be contained by the system under operation in order that the system may be used effectively. If there are different types of nodes in the system then it may be necessary to have at least a certain number of each type in the system. Ordinarily, it may be sufficient for the specified nodes to form a connected communications graph. If, instead, the basic system also specifies some necessary links between the nodes, then the problem of determining reusability is similar to the problem of Hayes [10] in that it becomes necessary to determine whether the reconfigured system contains a subgraph isomorphic to the basic graph for the system. For convenience, the latter type of reusability will be termed second order reusability in contrast to first order reusability where only the connectivity between the basic system nodes is important.

The following result may be proved without difficulty.

Theorem 6.2

If the basic system configuration requires n_i nodes of type i then a d-fault testable and reconfigurable system is first order reusable if and only if there are $d + n_i$ nodes of type i.

Proof:

If there are fewer than $d + n_i$ nodes of type i then the occurrence of d faults among the nodes of type i alone results in fewer than n_i nodes of type i remaining in the reconfigured system and the system is not reusable.

Conversely, if there are at least $d + n_i$ nodes in the system, then the fact that the system is d-fault reconfigurable implies that all the nodes in the system must remain connected even after d faults have occurred. The resulting system after reconfiguration is guaranteed to have at least n_i nodes of type i implying reusability. Q.E.D.

What the above result indicates is that the three graphs, viz., the system diagnostic graph, the system communications graph without distinguishing node types, and the system communications graph with node type information, have their own significance. Each of the graphs must be individually and systematically analyzed in order to draw meaningful conclusions about the behavior of the system under faults.

An observation about reconfigurability and reusability in actual systems is in order here. In all the foregoing discussion the tolerance to faults was determined by the worst possible case. In actual systems it may quite well be the case that a set of k faults which occur in a d-fault testable, reconfigurable and reusable system may actually leave a system which is better than (d-k)-fault testable, reconfigurable and

reusable. The maximum number of faults that the system can tolerate while remaining usable is a very optimistic measure and is not a judicious one in designing fault-tolerant systems except from the probabilistic point of view. The probabilistic treatment could take account of the fact that systems designed to be d-fault testable, reconfigurable and reusable are in a sense "over"-designed and can tolerate, with a high probability, a greater number of faults. An interesting treatment of this problem and the relation between the reliability and the performance of systems with time is presented in a paper by Abraham and Metze [23].

When the units involved in the operation of the system are non-homogeneous in nature, second order reusability must be considered. The redundancy in terms of the total number of extra units required increases in this case and the probability of the system remaining useful after a specified number of faults will tend to be higher. This is because the total number of redundant units in the system is considerably larger when there are many types of units than when the system is homogeneous. The deterministic aspects of this problem involve results from the graph theoretic problem of subgraph isomorphism [32] and will not be discussed here.

6.4 Self-Testable System Design

The results on the analysis of systems for testability, reconfigurability and reusability can be usefully employed in the design of nearly self-testable systems. Traditionally, the design of systems has been approached with an ultimate aim of obtaining "optimal" designs. The optimality criteria may include cost of the designed system, performance of the designed system, cost per performance, etc. The search for

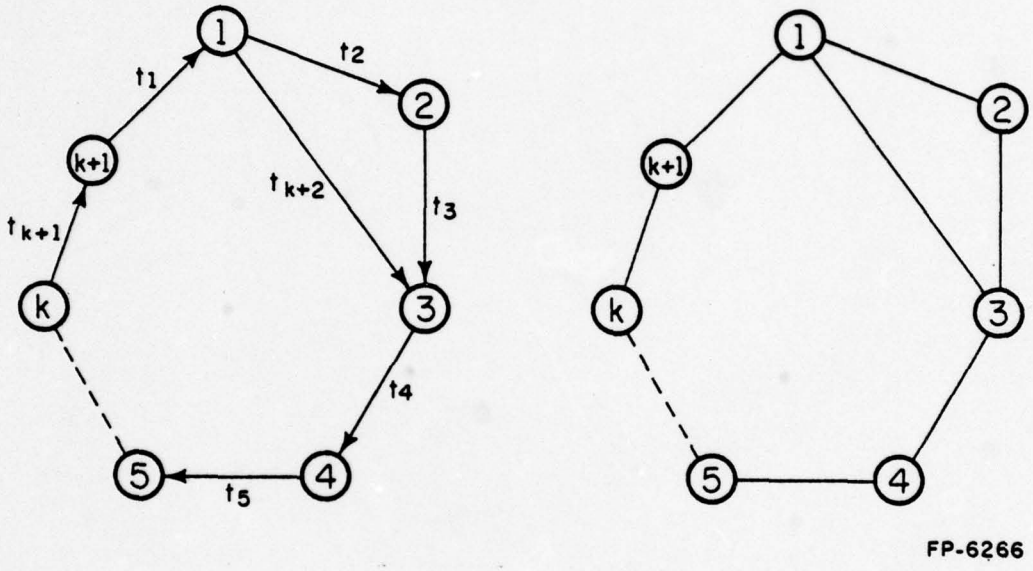
optimality is itself costly, however, and often an easily derived near-optimal design may be preferred to the effort required to derive an optimal design. This section will present some designs which are simple, easy to implement, and easily extensible.

Assume that it is required to design a 1-fault testable system consisting of just one type of node with the basic system requiring k nodes. Assuming that any node may be tested by any other node, the system whose graphs are shown in Figure 6.3 is a simple and almost optimal design.

Using the cooperative testing technique, the initial node 3 may be tested. Thereafter, node i tests node i + 1. The system is 1-fault reconfigurable because every node is contained in a Hamiltonian circuit of the system communications graph. By Theorem 6.2 since there are k + 1 nodes in the system and the system is 1-fault testable and reconfigurable, it must be reusable.

The particular system is near optimal if the criterion for optimality is the total number of nodes and links in the system. It may be easily shown that no less than k + 1 nodes and k + 1 links can meet the system requirements. The specified system has k + 1 nodes and k + 2 links.

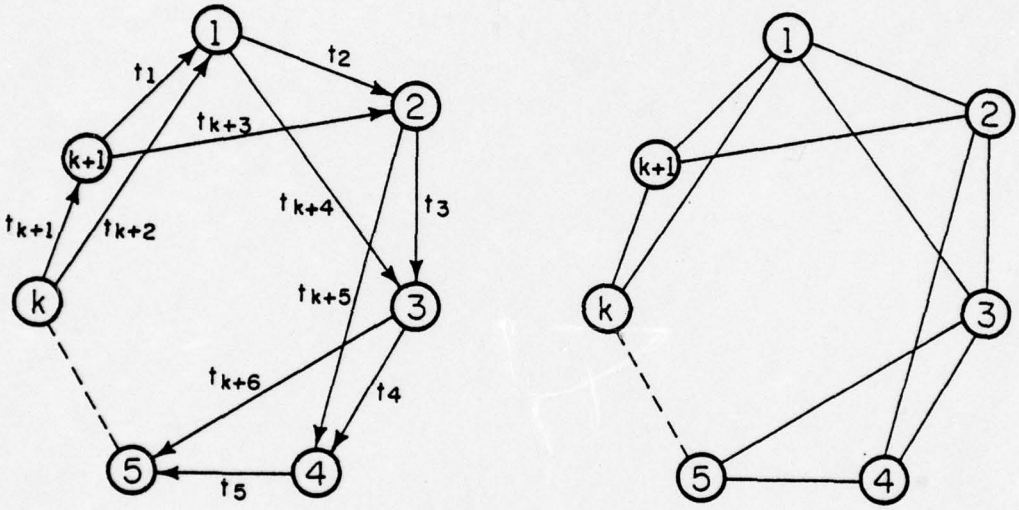
Considering the time taken for reconfiguration in the above system, it is seen that in the worst case, the information must travel over k - 1 nodes. This happens when some node i, $3 \leq i \leq k + 1$, finds a fault in the node that it tests. A design which is better from the point of view of reconfiguration delay is shown in Figure 6.4.



(a) Diagnostic Graph

(b) Communications Graph

Figure 6.3 A 1-Fault Testable System



(a) Diagnostic Graph

(b) Communications Graph

Figure 6.4 A 1-Fault Testable System with Low Reconfiguration Delay

The communications graph is symmetric and further the connectivity of the graph is 3. Hence by Theorem 6.1, the reconfiguration delay for the system is not greater than 4. The following result however indicates that the reconfiguration delay for the system is ≤ 2.

Theorem 6.3

In the system of Fig. 6.4, if the chosen roving graph is the directed Hamiltonian chain $1 \rightarrow 2 \rightarrow \ldots \rightarrow k + 1$ with node 1 as the initial node, then the reconfiguration delay of the system is ≤ 2.

Proof:

(All arithmetic is assumed modulo n.) If node i finds node i + 1 faulty then node i needs to inform node i - 1, node i + 2 and node i + 3. Node i is connected to the first two of these nodes, while node i + 3 may be reached through node i + 2 in 2 hops. Hence the theorem. Q.E.D.

When designing for d-fault systems, the same technique may be directly extended. Thus Figure 6.5 shows a homogeneous 3-fault testable system which is reconfigurable and reusable for a basic system of 7 nodes. The initial node, node 7, is tested by the cooperative testing technique using nodes 1 through 6.

The roving graph for the system is simply the graph of Figure 6.5 without the directed arcs to node 7. The nodes 8, 9 and 10 are the typical nodes for the system. If the basic system is required to have k nodes, $k > 7$, then it is sufficient to have the first seven nodes as shown, and the remaining k - 7 nodes connected as for nodes 8, 9 and 10. Thus expanding the system is straightforward.

The limitation that one encounters in extending this approach to d-fault testable systems is the number of ports that must exist in the

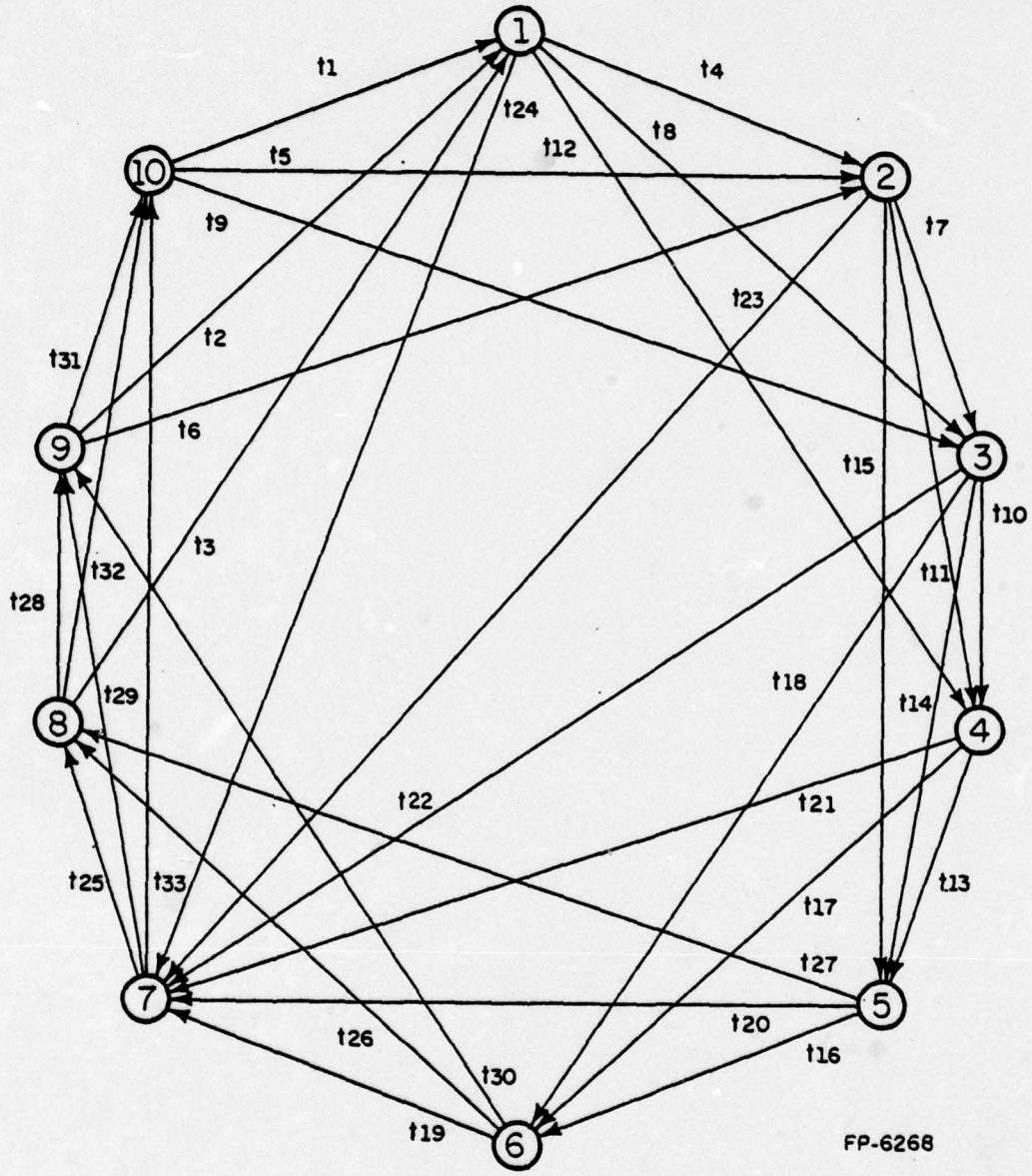


Figure 6.5 Diagnostic Graph for a 3-Fault Testable System

system nodes. This, in fact, is an inherent limitation of the roving diagnosis approach just as it was in the case of one-step diagnosability or sequential diagnosability without repair, and is the penalty that is paid for the attempt to eliminate global supervision.

This section is concluded with a design of a system in which the individual nodes cannot be tested by one unit alone. Assume that the system is to be 1-fault detectable, reconfigurable and reusable with each unit requiring 3 other units to perform a complete test on it. (This may be the case when the individual units are too complex to be tested by just one other unit of its type or when resource limitations prevent a complete test by one unit.)

The roving graph for the system, assuming a basic system requiring 9 homogeneous nodes, is shown in Figure 6.6. Except for the fact that there are 3 initial nodes instead of one, the graph is similar to the roving graph for the 3-fault testable system, when labels on the test edges are removed.

The most convenient way to perform the diagnosis of initial nodes would be to employ 6 nodes to diagnose unit 1, 5 to diagnose unit 2 and 4 to diagnose unit 3 as shown in Figure 6.7. While this does make initial node diagnosis more expensive than in the case of single mask per test systems, it is better than using six nodes to diagnose each of the nodes 1, 2 and 3. The communications graph must contain links between the nodes 4 through 9, as before.

A proof that the graphs of Figures 6.6 and 6.7 achieve their intended purpose is straightforward and is not presented here.

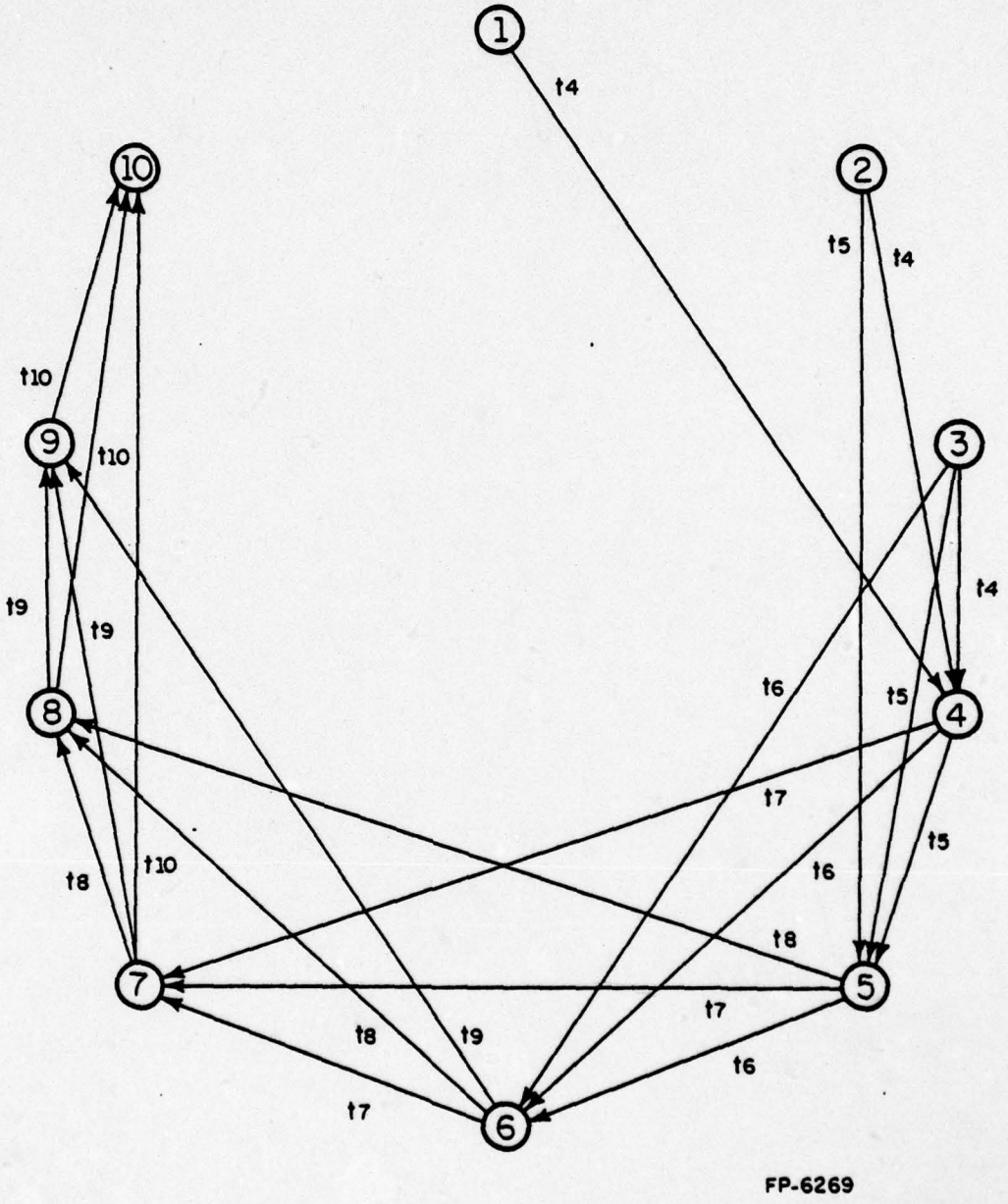


Figure 6.6 Roving Graph for a 1-Fault Testable System with 3 Masks Per Test

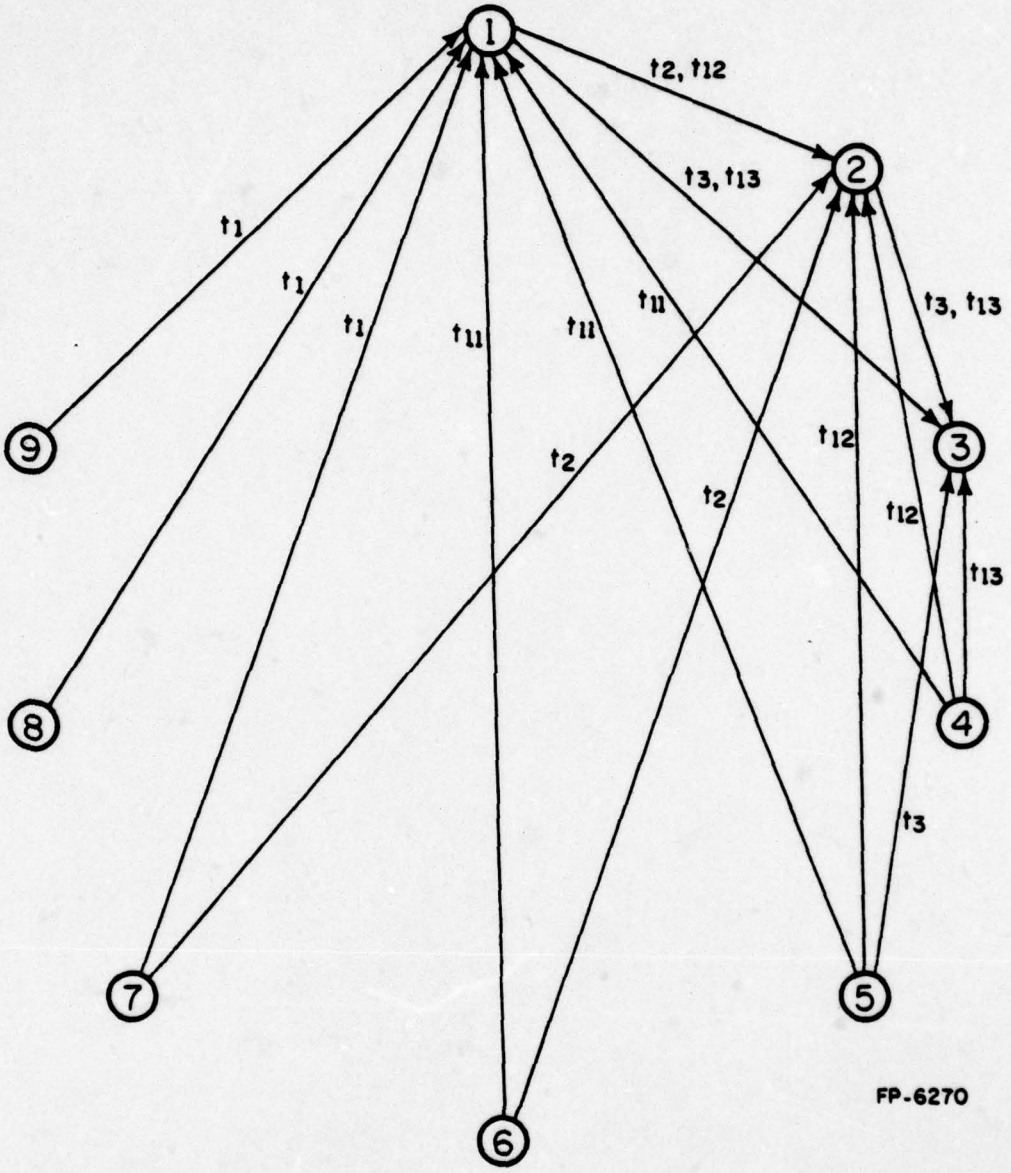
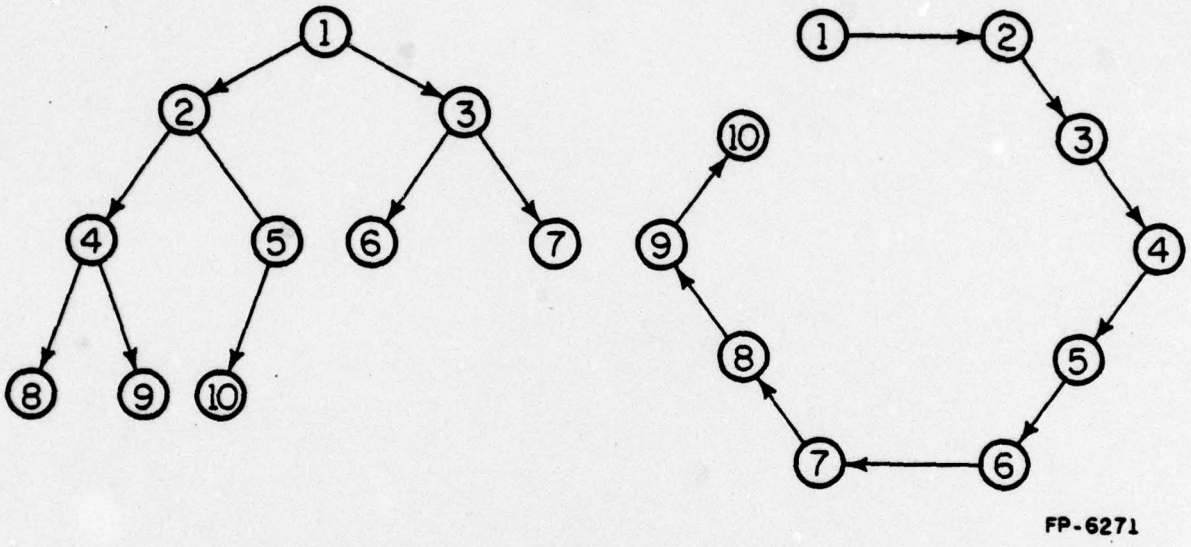


Figure 6.7 Diagnosis of Initial Nodes for the System of Figure 6.6

6.5 Miscellaneous Practical Aspects

One of the problems that is likely to be encountered in practical implementations of the roving diagnosis scheme is the actual communication between the various nodes during testing. For example, during testing of one node by another, the faulty node being tested may simply "hang" and not send any response to test stimuli. The testing node, of course, must be able to recognize such a condition from a normal delay over communication links. It is under these circumstances that the value of a testing link directly connecting the testing node and the tested node is realized. The testing node can now simply wait for a prespecified length of time before declaring the node being tested as faulty or slower than acceptable.

A related problem is one of minimization of the number of hops over which messages must travel during the testing itself. Consider the two roving graphs for a homogeneous system shown in Figure 6.8. The advantage in having a regular balanced tree over a chain (which is also a tree, but a degenerate one) is that testing can proceed in parallel thus minimizing the total time involved in testing. In the case of the chain, however, there is no choice as to the sequence in which the testing must proceed. Clearly, a node that has just been found to be fault-free will proceed to act as the next tester and thus a true roving is achieved. For the tree, one possible sequence starts with 1 testing 2, 2 testing 4, 4 testing 8 and a jump back to 1 to test 3. In order for this to happen, a message must be sent back from node 8 to node 1 which involves 3 hops. Thus, while such a sequence does not tie up any node for too long in diagnosis, it is not very efficient from the point of view of total number of message hops.



(a) A Tree

(b) A Chain

Figure 6.8 Two Possible Roving Graphs for a 10-Node Homogeneous System

A better sequence which can exploit the parallelism in the system also would start with 1 testing 2 and followed by 1 testing 3 in conjunction with 2 testing 4, and so on. This assumes that more than one area can be performing diagnosis at the same time. If because of some performance constraints only one area should be diagnosing at any given time then it can be easily seen that the chain would be the best roving graph in terms of minimizing the number of messages sent. In SMPT diagnostic graphs, this implies the necessity for the existence of a Hamiltonian path. (The roving graph for the 3 fault testable system of Figure 6.5 satisfies this condition. It further enjoys the advantage of possessing a Hamiltonian path even if the system reconfigures around 1 or 2 faults.)

The final problem to be discussed involves the propagation of the network topology information to all the nodes in the system. The primary question here is whether all the nodes in the system are required to know the complete topology of the system in order to function effectively or whether the knowledge of who its active neighbors are is sufficient for a node.

This question assumes importance during reconfiguration because, except in very few topologies like the complete graph, the greater the number of nodes which need to be informed about a fault, the greater will be the time required to effect reconfiguration.

Some guidelines for design have already been presented in Section 6.2 for the case when only the neighbors of a faulty node need to be informed about the fault.

The distance between two nodes f_i and f_j in a connected graph is defined as the minimum number of edges contained in a chain (a directed chain if the graph is a directed graph) joining f_i and f_j. The maximum distance between all pairs of vertices in a connected graph is called the diameter of the graph. Thus in order to reduce reconfiguration delay, when all the nodes need to be informed about a fault in some node, the diameter of the communications graph must be reduced. If all the nodes in the system know the system topology completely at the beginning, then on the occurrence of a fault, the simple propagation of the fault information down the shortest paths will ensure a complete update of the topology at each node.

If, on the other hand, each node in the system needs to know only the distances between itself and the other nodes, a simple ARPANET-like strategy [33] may be used. Here each node gathers information from each of its neighbors to determine the distances between them and all the other nodes. By comparing the information from the neighbors, the node can decide the shortest distance between itself and the rest of the nodes in the system. It can also decide which specific neighbor is on the shortest path between it and some other node.

One of the major advantages of the roving diagnosis scheme and the associated models as presented here is the fact that they can be applied to either loosely coupled distributed computer communication networks or to tightly coupled computer systems. The details of the demarcation into convenient functional blocks may vary in the two cases, but once the relevant graphs have been drawn, the possibility of roving diagnosis may be investigated through the methods described. Further,

the theory developed can also be used to determine suitable modifications to an existing system so that the diagnosis of the system may be performed with virtual elimination of a global supervisor. (It may be recalled that the only hard-core required for the presented approach is a mechanism by which the testing units for the initial node are instructed to begin testing the initial node.)

7. CONCLUDING REMARKS

This chapter concludes the thesis by summarizing the salient results that have been obtained and discussing directions in which further work may be done.

7.1 Detection of a Fault-Free Unit

As shown in Chapter 2, the problems of finding at least one good unit in the system is identical to d/s diagnosability of Friedman [16] for the special case of $s = n - 1$, n being the total number of units in the system. The necessary and sufficient conditions for d/(n-1) diagnosability were derived. In the case when no more than one unit is required to test another, it was shown that d-fault diagnosability with repair is a sufficient condition for finding at least one good unit in the system if the system diagnostic graph is connected, while it is also a necessary condition if the graph is strongly connected. In most other cases however d/(n-1) diagnosability seems to be more easily satisfiable in systems than d-fault diagnosability with repair (and hence also easier to satisfy than d-fault diagnosability without repair). A graph-theoretic parameter called the system implication index was defined. It was shown that for a system in which every unit has exactly one test (possibly involving many diagnosing units), d/(n-1) diagnosability may be ensured by making the implication index large enough.

7.2 Decomposition of Systems

Chapter 3 related the closure index, and hence the diagnosability, of a system with the closure indices and the diagnosabilities of the composing subsystems for the system. While convenient upper and lower

bounds on the diagnosability of the system may be obtained, an exact value of the diagnosability often requires more detailed knowledge about the subsystems themselves. The knowledge of the characteristics of the interconnections, for example its directionality or whether it is constructive or not, is further useful information in obtaining better bounds on the system diagnosability.

The open-circuit diagnosability defined in the chapter is a useful tool in system analysis. It is felt that there should exist at least one other measure, which, along with open-circuit diagnosability, will characterize a subsystem completely with respect to its points of connection to the outside world. Further work needs to be done in this direction.

7.3 Roving Diagnosis

Roving diagnosis, described in Chapter 4, is shown to be a practical, feasible means of diagnosing large systems, with almost no aid of an external supervisor. The key requirements for a system to be able to perform roving diagnosis are the existence of a suitable roving subgraph in the system diagnostic graph and a mechanism to test certain distinguished nodes called the initial nodes. Chapter 5 discussed the time required to "rove" through the system and provided algorithms for determining appropriate roving graphs given the fault model and the system diagnostic graph.

Reconfiguration of the system under a fault and reusability of the system after reconfiguration were issues discussed in Chapter 6. The problem of reconfigurability has been related to the well studied graph-theoretic problem of connectivity. It was also shown that certain

properties of the system communications graph lead to desirable systems where the delay in reconfiguring after a fault is small. Reusability of systems, especially when different types of nodes are involved, is a more difficult issue. At present, results are sparse. More work may be done in this area. A relationship may exist between the reusability problem and the fault recovery problem studied by Hayes and Yanney [10,21].

As evident from the sample designs of Chapter 6, the complexity of interconnections increases rapidly as the system is required to tolerate more faults. It must be noted, nevertheless, that this complexity, while worse in comparison to sequential diagnosability, is comparable to the complexity for one-step diagnosability, and probably lower in comparison to intermittent fault diagnosability. Intuitively, one may attribute this as the cost for being able to locate all the faults in the system up to a given maximum without ambiguity and without analyzing the entire system fault syndrome. Added to that are the benefits due to virtual elimination of global hard-core and due to the parallelism of diagnosis with other computation.

More fruitful work remains to be done, especially in the area of probabilistic analysis of roving diagnosis. A start in this direction was made by Abraham and Metze [23], where an example is provided demonstrating that roving diagnosis leads to systems which are far superior in performance while being almost as reliable compared to systems using classical redundancy. Further examples need to be worked out especially for cases involving non-homogeneous nodes and reusability of the second order. A probabilistic analysis of the various schemes for testing initial nodes would also be quite useful.

As mentioned in Chapter 1, system diagnosis assumes that the details of diagnosis at the lower component level can be taken care of separately. For large LSI system modules like microprocessors, the efficient generation of test-sets with good fault coverage is still an open problem. Self-checking system modules, which unfortunately have not gained popularity with commercial manufacturers yet, can be used to great advantage because the built-in checking mechanism allows one to prolong the periods between tests. (All the results of Chapters 5 and 6 are valid if the modules are self-checking, provided that the diagnostic graph reflects the fact that any test for a fault in a unit is invalidated also by a fault in the unit monitoring the checker output of the former unit.)

The recent flurry of activity in the areas of self-checking and LSI testing is hence very encouraging.

REFERENCES

[1] F. P. Preparata, G. Metze, and R. T. Chien, "On the connection assignment problem of diagnosable systems," IEEE Trans. Electron. Comput., vol. EC-16, pp. 848-854, Dec. 1967.

[2] F. P. Preparata, "Some results on sequentially diagnosable systems," in Proc. Hawaii Int. Conf. Syst. Sci., University of Hawaii Press, Honolulu, Hawaii, Jan. 1968, pp. 623-626.

[3] S. L. Hakimi and A. T. Amin, "Characterization of connection assignment of diagnosable systems," IEEE Trans. Comput., vol. C-23, pp. 86-88, Jan. 1974.

[4] J. D. Russell and C. R. Kime, "System fault diagnosis: closure and diagnosability with repair," IEEE Trans. Comput., vol. C-24, pp. 1079-1088, Nov. 1975.

[5] J. D. Russell and C. R. Kime, "System fault diagnosis: masking, exposure, and diagnosability without repair," IEEE Trans. Comput., vol. C-24, pp. 1155-1161, Dec. 1975.

[6] S. N. Maheshwari and S. L. Hakimi, "On models for diagnosable systems and probabilistic fault diagnosis," IEEE Trans. Comput., vol. C-25, pp. 228-236, Mar. 1976.

[7] H. Fujiwara and K. Kinoshita, "Connection assignments for probabilistically diagnosable systems," IEEE Trans. Comput., vol. C-27, pp. 280-283, Mar. 1978.

[8] S. Mallela and G. M. Masson, "Diagnosable systems for intermittent faults," IEEE Trans. Comput., vol. C-27, pp. 560-566, June 1978.

[9] F. Barsi, F. Grandoni and P. Maestrini, "A theory of diagnosability of digital systems," IEEE Trans. Comput., vol. C-25, pp. 585-593, June 1976.

[10] J. P. Hayes, "A graph model for fault-tolerant computing systems," IEEE Trans. Comput., vol. C-25, pp. 875-884, Sept. 1976.

[11] C. V. Ramamoorthy, "A structural theory of machine diagnosis," in Proc. 1967 Spring Joint Comput. Conf., AFIPS Conf. Proc., vol. 30, Washington, D.C.: Thompson, 1967, pp. 743-756.

[12] C. V. Ramamoorthy and L.-C. Chang, "System segmentation for parallel diagnosis of computers," IEEE Trans. Comput., vol. C-20, pp. 261-270, Mar. 1971.

[13] C. V. Ramamoorthy and W. Mayeda, "Computer diagnosis using the blocking gate approach," IEEE Trans. Comput., vol. C-20, pp. 1294-1299, Nov. 1971.

[14] R. P. Batni and C. R. Kime, "A module-level testing approach for combinational networks," IEEE Trans. Comput., vol. C-25, pp. 594-604, June 1976.

[15] R. A. Poisel and C. R. Kime, "A system interconnection model for diagnosability analysis," in Proc. 1977 Int. Symp. Fault-Tolerant Computing, IEEE Computer Society Publications, pp. 59-63, 1977.

[16] A. D. Friedman, "A new measure of digital system diagnosis," in Dig. 1975 Int. Symp. Fault-Tolerant Computing, IEEE Computer Society Publications, pp. 167-170, 1975.

[17] J. P. Roth, "Algebraic topological methods for the synthesis of switching systems, I," Trans. Amer. Math. Soc., vol. 88, pp. 301-326, July 1958.

[18] S. Karunanithi and A. D. Friedman, "System diagnosis with t/s diagnosability," in Proc. 1977 Int. Symp. Fault-Tolerant Computing, IEEE Computer Society Publications, pp. 65-71, 1977.

[19] J. E. Smith, "Universal system diagnosis algorithms," unpublished document, 1977.

[20] J. A. McPherson and C. R. Kime, "A two-level approach to modeling system diagnosability," Proc. 1976 Int. Symp. Fault-Tolerant Computing, IEEE Computer Society Publications, pp. 33-38, 1976.

[21] J. P. Hayes and R. Yanney, "Fault recovery in multiprocessor networks," Digest of Papers, 1978 Fault-Tolerant Computing, IEEE Computer Society Publications, pp. 123-128, 1978.

[22] J. M. McQuillan, "Design considerations for routing algorithms in computing networks," in Proc. Hawaii Int. Conf. Syst. Sci., University of Hawaii Press, Honolulu, Hawaii, Jan. 1974.

[23] J. A. Abraham and G. Metze, "Roving diagnosis for high performance digital systems," in Proc. Conf. on Info. Sci. and Syst., The Johns Hopkins University, Baltimore, Maryland, March 1978, pp. 221-226.

[24] R. G. Busacker and T. L. Saaty, Finite Graphs and Networks: An introduction with applications, New York: McGraw Hill, 1965.

[25] G. H. Danielson, "On finding the simple paths and circuits in a graph," IEEE Trans. Circuit Theory, vol. CT-15, pp. 294-295, September 1968.

[26] S. R. Petrick, "On the minimization of Boolean functions," Proc. Symp. on Switching Theory, ICIP, Paris, June 1959.

[27] W. C. Carter and P. R. Schneider, "Design of dynamically checked computers," IFIP 68, vol. 2, Edinburgh, Scotland, August 1968, pp. 878-883.

[28] D. A. Anderson, "Design of self-checking digital networks using coding techniques," Coordinated Science Laboratory Report R-527, University of Illinois, Sept. 1971.

[29] J. E. Smith and G. Metze, "The design of totally self-checking combinational circuits," Proc. 1977 Int. Symp. Fault-Tolerant Computing, IEEE Computer Society Publications, pp. 130-134, 1977.

[30] R. S. Wilkov, "Analysis and design of reliable computer networks," IEEE Trans. Communications, vol. COM-20, pp. 660-678, June 1972.

[31] H. Frank and I. T. Frisch, Communication, Transmission, and Transportation Networks, Reading, Mass., Addison-Wesley, 1971.

[32] A. T. Berztiss, "A backtrack procedure for isomorphism of directed graphs," JACM, vol. 20, pp. 365-377, 1973.

[33] G. L. Fultz and L. Kleinrock, "Adaptive routing techniques for store-and-forward computer-communication networks," Proc. 1971 IEEE Int. Conf. Communications, vol. 7, pp. 3901-3908.

VITA

Ravindra Kumar Nair was born in Madras, India on May 7, 1953. He received a B. Tech degree in Electronics and Electrical Communication Engineering from the Indian Institute of Technology, Kharagpur in 1974 and an M.S. degree in Computer Science from the University of Illinois at Urbana-Champaign in 1976. He was a teaching and research assistant with the Department of Computer Science from 1974 to 1976 and a research assistant with the Fault-tolerant Systems and Computer Architecture group at the Coordinated Science Laboratory from 1976 to 1978.